Generation of finite groups

with applications to computing normalisers

Colva Roney-Dougal
University of St Andrews

11th October 2017

Minimal generation

Minimal generation

$d(G)$ – minimum number of generators of a (finite) group G.

Minimal generation

$d(G)$ – minimum number of generators of a (finite) group G.

Theorem (Consequence of CFSG)

Let G be a finite simple group.

Minimal generation

$d(G)$ – minimum number of generators of a (finite) group G.

Theorem (Consequence of CFSG)

Let G be a finite simple group. Then $d(G) \leq 2$.

Minimal generation

$d(G)$ – minimum number of generators of a (finite) group G.

Theorem (Consequence of CFSG)

Let G be a finite simple group. Then $d(G) \leq 2$.

G is almost simple if there exists a nonabelian simple S s.t. $S \trianglelefteq G \leq \mathrm{Aut}(S)$.

Minimal generation

$d(G)$ – minimum number of generators of a (finite) group G.

Theorem (Consequence of CFSG)

Let G be a finite simple group. Then $d(G) \leq 2$.

G is almost simple if there exists a nonabelian simple S s.t. $S \trianglelefteq G \leq \mathrm{Aut}(S)$. $S = \mathrm{Soc}(G)$ is the socle of G.

Minimal generation

$d(G)$ – minimum number of generators of a (finite) group G.

Theorem (Consequence of CFSG)

Let G be a finite simple group. Then $d(G) \leq 2$.

G is almost simple if there exists a nonabelian simple S s.t. $S \trianglelefteq G \leq \mathrm{Aut}(S)$. $S = \mathrm{Soc}(G)$ is the socle of G.

Theorem (Dalla Volta & Lucchini 95)

Let G be a finite almost simple group with socle S. Then $d(G) \leq 3$

Minimal generation

$d(G)$ – minimum number of generators of a (finite) group G.

Theorem (Consequence of CFSG)

Let G be a finite simple group. Then $d(G) \leq 2$.

G is almost simple if there exists a nonabelian simple S s.t. $S \trianglelefteq G \leq \mathrm{Aut}(S)$. $S = \mathrm{Soc}(G)$ is the socle of G.

Theorem (Dalla Volta & Lucchini 95)

Let G be a finite almost simple group with socle S. Then $d(G) \leq 3$, and $d(G) = 3$ if and only if $d(G/S) = 3$.

Minimal generation

$d(G)$ – minimum number of generators of a (finite) group G.

Theorem (Consequence of CFSG)

Let G be a finite simple group. Then $d(G) \leq 2$.

G is almost simple if there exists a nonabelian simple S s.t. $S \trianglelefteq G \leq \mathrm{Aut}(S)$. $S = \mathrm{Soc}(G)$ is the socle of G.

Theorem (Dalla Volta & Lucchini 95)

Let G be a finite almost simple group with socle S. Then $d(G) \leq 3$, and $d(G) = 3$ if and only if $d(G/S) = 3$.

Theorem (Burness, Liebeck & Shalev 13)

G – almost simple with socle S. H – maximal subgroup of S.

Minimal generation

$d(G)$ – minimum number of generators of a (finite) group G.

Theorem (Consequence of CFSG)

Let G be a finite simple group. Then $d(G) \leq 2$.

G is almost simple if there exists a nonabelian simple S s.t. $S \trianglelefteq G \leq \mathrm{Aut}(S)$. $S = \mathrm{Soc}(G)$ is the socle of G.

Theorem (Dalla Volta & Lucchini 95)

Let G be a finite almost simple group with socle S. Then $d(G) \leq 3$, and $d(G) = 3$ if and only if $d(G/S) = 3$.

Theorem (Burness, Liebeck & Shalev 13)

G – almost simple with socle S. H – maximal subgroup of S. Then $d(H) \leq 4$.

Minimal generation

$d(G)$ – minimum number of generators of a (finite) group G.

Theorem (Consequence of CFSG)

Let G be a finite simple group. Then $d(G) \leq 2$.

G is almost simple if there exists a nonabelian simple S s.t. $S \trianglelefteq G \leq \mathrm{Aut}(S)$. $S = \mathrm{Soc}(G)$ is the socle of G.

Theorem (Dalla Volta & Lucchini 95)

Let G be a finite almost simple group with socle S. Then $d(G) \leq 3$, and $d(G) = 3$ if and only if $d(G/S) = 3$.

Theorem (Burness, Liebeck & Shalev 13)

G – almost simple with socle S. H – maximal subgroup of S. Then $d(H) \leq 4$.
M – maximal subgroup of G.

Minimal generation

$d(G)$ – minimum number of generators of a (finite) group G.

Theorem (Consequence of CFSG)

Let G be a finite simple group. Then $d(G) \leq 2$.

G is almost simple if there exists a nonabelian simple S s.t. $S \trianglelefteq G \leq \mathrm{Aut}(S)$. $S = \mathrm{Soc}(G)$ is the socle of G.

Theorem (Dalla Volta & Lucchini 95)

Let G be a finite almost simple group with socle S. Then $d(G) \leq 3$, and $d(G) = 3$ if and only if $d(G/S) = 3$.

Theorem (Burness, Liebeck & Shalev 13)

G – almost simple with socle S. H – maximal subgroup of S. Then $d(H) \leq 4$.
M – maximal subgroup of G. Then $d(M) \leq 6$.

Minimal generation

$d(G)$ – minimum number of generators of a (finite) group G.

Theorem (Consequence of CFSG)

Let G be a finite simple group. Then $d(G) \leq 2$.

G is almost simple if there exists a nonabelian simple S s.t. $S \trianglelefteq G \leq \mathrm{Aut}(S)$. $S = \mathrm{Soc}(G)$ is the socle of G.

Theorem (Dalla Volta & Lucchini 95)

Let G be a finite almost simple group with socle S. Then $d(G) \leq 3$, and $d(G) = 3$ if and only if $d(G/S) = 3$.

Theorem (Burness, Liebeck & Shalev 13)

G – almost simple with socle S. H – maximal subgroup of S. Then $d(H) \leq 4$.
M – maximal subgroup of G. Then $d(M) \leq 6$.

Probabilistic generation of simple groups

Probabilistic generation of simple groups

$P_G(k)$ – probability that k random elts of G generate G.

Probabilistic generation of simple groups

$P_G(k)$ – probability that k random elts of G generate G.

Theorem (Dixon 69; Kantor & Lubotzky 90; Liebeck & Shalev 95)

S – finite simple group.

Probabilistic generation of simple groups

$P_G(k)$ – probability that k random elts of G generate G.

Theorem (Dixon 69; Kantor & Lubotzky 90; Liebeck & Shalev 95)

S – finite simple group. Then $P_S(2) \to 1$ as $|S| \to \infty$.

Probabilistic generation of simple groups

$P_G(k)$ – probability that k random elts of G generate G.

Theorem (Dixon 69; Kantor & Lubotzky 90; Liebeck & Shalev 95)

S – finite simple group. Then $P_S(2) \to 1$ as $|S| \to \infty$.

Theorem (Menezes, Quick, CMRD 13)

S – finite simple group.

Probabilistic generation of simple groups

$P_G(k)$ – probability that k random elts of G generate G.

Theorem (Dixon 69; Kantor & Lubotzky 90; Liebeck & Shalev 95)

S – finite simple group. Then $P_S(2) \to 1$ as $|S| \to \infty$.

Theorem (Menezes, Quick, CMRD 13)

S – finite simple group. Then $P_S(2) \geq 53/90$

Probabilistic generation of simple groups

$P_G(k)$ – probability that k random elts of G generate G.

Theorem (Dixon 69; Kantor & Lubotzky 90; Liebeck & Shalev 95)

S – finite simple group. Then $P_S(2) \to 1$ as $|S| \to \infty$.

Theorem (Menezes, Quick, CMRD 13)

S – finite simple group. Then $P_S(2) \geq 53/90$, with equality if and only if $S = \mathrm{A}_6$.

Probabilistic generation of simple groups

$P_G(k)$ – probability that k random elts of G generate G.

Theorem (Dixon 69; Kantor & Lubotzky 90; Liebeck & Shalev 95)

S – finite simple group. Then $P_S(2) \to 1$ as $|S| \to \infty$.

Theorem (Menezes, Quick, CMRD 13)

S – finite simple group. Then $P_S(2) \geq 53/90$, with equality if and only if $S = \mathrm{A}_6$.

$m(G)$ – minimal index of a proper subgroup of G.

Probabilistic generation of simple groups

$P_G(k)$ – probability that k random elts of G generate G.

Theorem (Dixon 69; Kantor & Lubotzky 90; Liebeck & Shalev 95)

S – finite simple group. Then $P_S(2) \to 1$ as $|S| \to \infty$.

Theorem (Menezes, Quick, CMRD 13)

S – finite simple group. Then $P_S(2) \geq 53/90$, with equality if and only if $S = \mathrm{A}_6$.

$m(G)$ – minimal index of a proper subgroup of G.

Theorem (Liebeck & Shalev 96)

There exist constants α and β

Probabilistic generation of simple groups

$P_G(k)$ – probability that k random elts of G generate G.

Theorem (Dixon 69; Kantor & Lubotzky 90; Liebeck & Shalev 95)

S – finite simple group. Then $P_S(2) \to 1$ as $|S| \to \infty$.

Theorem (Menezes, Quick, CMRD 13)

S – finite simple group. Then $P_S(2) \geq 53/90$, with equality if and only if $S = \mathrm{A}_6$.

$m(G)$ – minimal index of a proper subgroup of G.

Theorem (Liebeck & Shalev 96)

There exist constants α and β s.t. for all finite simple groups S,

Probabilistic generation of simple groups

$P_G(k)$ – probability that k random elts of G generate G.

Theorem (Dixon 69; Kantor & Lubotzky 90; Liebeck & Shalev 95)

S – finite simple group. Then $P_S(2) \to 1$ as $|S| \to \infty$.

Theorem (Menezes, Quick, CMRD 13)

S – finite simple group. Then $P_S(2) \geq 53/90$, with equality if and only if $S = \mathrm{A}_6$.

$m(G)$ – minimal index of a proper subgroup of G.

Theorem (Liebeck & Shalev 96)

There exist constants α and β s.t. for all finite simple groups S,

$$1 - \frac{\alpha}{m(S)} < P_S(2) < 1 - \frac{\beta}{m(S)}.$$

Probabilistic generation of simple groups

$P_G(k)$ – probability that k random elts of G generate G.

Theorem (Dixon 69; Kantor & Lubotzky 90; Liebeck & Shalev 95)

S – finite simple group. Then $P_S(2) \to 1$ as $|S| \to \infty$.

Theorem (Menezes, Quick, CMRD 13)

S – finite simple group. Then $P_S(2) \geq 53/90$, with equality if and only if $S = \mathrm{A}_6$.

$m(G)$ – minimal index of a proper subgroup of G.

Theorem (Liebeck & Shalev 96)

There exist constants α and β s.t. for all finite simple groups S,

$$1 - \frac{\alpha}{m(S)} < P_S(2) < 1 - \frac{\beta}{m(S)}.$$

Minimal generation of permutation groups

Minimal generation of permutation groups

$G \leq S_n$ is transitive

Minimal generation of permutation groups

$G \leq S_n$ is transitive if for all $\alpha, \beta \in \{1, \ldots, n\}$ there exists $g \in G$ s.t. $\alpha^g = \beta$.

Minimal generation of permutation groups

$G \leq \mathsf{S}_n$ is transitive if for all $\alpha, \beta \in \{1, \ldots, n\}$ there exists $g \in G$ s.t. $\alpha^g = \beta$.

Theorem (Cameron, Solomon & Turull 89; Neumann)

Let $G \leq \mathsf{S}_n$. Then $d(G) \leq \max\{n/2, 2\}$.

Minimal generation of permutation groups

$G \leq \mathsf{S}_n$ is transitive if for all $\alpha, \beta \in \{1, \ldots, n\}$ there exists $g \in G$ s.t. $\alpha^g = \beta$.

Theorem (Cameron, Solomon & Turull 89; Neumann)

Let $G \leq \mathsf{S}_n$. Then $d(G) \leq \max\{n/2, 2\}$.

Bound is best possible:

- If n is even then $C_2^{n/2} \leq \mathsf{S}_n$, and $d(C_2^{n/2}) = n/2$.

Minimal generation of permutation groups

$G \leq \mathsf{S}_n$ is transitive if for all $\alpha, \beta \in \{1, \ldots, n\}$ there exists $g \in G$ s.t. $\alpha^g = \beta$.

Theorem (Cameron, Solomon & Turull 89; Neumann)

Let $G \leq \mathsf{S}_n$. Then $d(G) \leq \max\{n/2, 2\}$.

Bound is best possible:

- If n is even then $C_2^{n/2} \leq \mathsf{S}_n$, and $d(C_2^{n/2}) = n/2$.
- $d(\mathsf{S}_3) = 2$.

Key ingredient of proof is:

Lemma (Wielandt)

Let $P \leq \mathsf{S}_{p^m}$ be a transitive p-group.

Minimal generation of permutation groups

$G \leq S_n$ is transitive if for all $\alpha, \beta \in \{1, \ldots, n\}$ there exists $g \in G$ s.t. $\alpha^g = \beta$.

Theorem (Cameron, Solomon & Turull 89; Neumann)

Let $G \leq S_n$. Then $d(G) \leq \max\{n/2, 2\}$.

Bound is best possible:

- If n is even then $C_2^{n/2} \leq S_n$, and $d(C_2^{n/2}) = n/2$.
- $d(S_3) = 2$.

Key ingredient of proof is:

Lemma (Wielandt)

Let $P \leq S_{p^m}$ be a transitive p-group. Then $d(P) \leq 1 + \sum_{i=0}^{m-2} p^i$.

Minimal generation of permutation groups

$G \leq S_n$ is transitive if for all $\alpha, \beta \in \{1, \ldots, n\}$ there exists $g \in G$ s.t. $\alpha^g = \beta$.

Theorem (Cameron, Solomon & Turull 89; Neumann)

Let $G \leq S_n$. Then $d(G) \leq \max\{n/2, 2\}$.

Bound is best possible:

- If n is even then $C_2^{n/2} \leq S_n$, and $d(C_2^{n/2}) = n/2$.
- $d(S_3) = 2$.

Key ingredient of proof is:

Lemma (Wielandt)

Let $P \leq S_{p^m}$ be a transitive p-group. Then $d(P) \leq 1 + \sum_{i=0}^{m-2} p^i$.

Corollary

If $P \leq S_n$ is a p-group, then $d(P) \leq n/2$.

Minimal generation of permutation groups

$G \leq S_n$ is transitive if for all $\alpha, \beta \in \{1, \ldots, n\}$ there exists $g \in G$ s.t. $\alpha^g = \beta$.

Theorem (Cameron, Solomon & Turull 89; Neumann)

Let $G \leq S_n$. Then $d(G) \leq \max\{n/2, 2\}$.

Bound is best possible:

- If n is even then $C_2^{n/2} \leq S_n$, and $d(C_2^{n/2}) = n/2$.
- $d(S_3) = 2$.

Key ingredient of proof is:

Lemma (Wielandt)

Let $P \leq S_{p^m}$ be a transitive p-group. Then $d(P) \leq 1 + \sum_{i=0}^{m-2} p^i$.

Corollary

If $P \leq S_n$ is a p-group, then $d(P) \leq n/2$.

Minimal generation of transitive groups

Minimal generation of transitive groups

Theorem (Cameron Solomon Turull; Neumann 89)

If $G \leq S_n$ is transitive

Minimal generation of transitive groups

Theorem (Cameron Solomon Turull; Neumann 89)

If $G \leq S_n$ *is transitive,* $n > 4$

Minimal generation of transitive groups

Theorem (Cameron Solomon Turull; Neumann 89)

If $G \leq S_n$ *is transitive,* $n > 4$ *and* $(G, n) \neq (D_8 \circ D_8, 8)$

Minimal generation of transitive groups

Theorem (Cameron Solomon Turull; Neumann 89)

If $G \leq S_n$ is transitive, $n > 4$ and $(G, n) \neq (D_8 \circ D_8, 8)$ then $d(G) < n/2$.

Minimal generation of transitive groups

Theorem (Cameron Solomon Turull; Neumann 89)

If $G \leq S_n$ is transitive, $n > 4$ and $(G, n) \neq (D_8 \circ D_8, 8)$ then $d(G) < n/2$.

Theorem (Lucchini, Menegazzo, Morigi 00)

There exists a constant c s.t. if $G \leq S_n$ is transitive

Minimal generation of transitive groups

Theorem (Cameron Solomon Turull; Neumann 89)

If $G \leq S_n$ is transitive, $n > 4$ and $(G, n) \neq (D_8 \circ D_8, 8)$ then $d(G) < n/2$.

Theorem (Lucchini, Menegazzo, Morigi 00)

There exists a constant c s.t. if $G \leq S_n$ is transitive, then $d(G) \leq cn/\sqrt{\log n}$.

Minimal generation of transitive groups

Theorem (Cameron Solomon Turull; Neumann 89)

If $G \leq S_n$ is transitive, $n > 4$ and $(G, n) \neq (D_8 \circ D_8, 8)$ then $d(G) < n/2$.

Theorem (Lucchini, Menegazzo, Morigi 00)

There exists a constant c s.t. if $G \leq S_n$ is transitive, then $d(G) \leq cn/\sqrt{\log n}$.

- Kovacs and Newman: for each prime p there exists a constant c_p

Minimal generation of transitive groups

Theorem (Cameron Solomon Turull; Neumann 89)

If $G \leq S_n$ is transitive, $n > 4$ and $(G, n) \neq (D_8 \circ D_8, 8)$ then $d(G) < n/2$.

Theorem (Lucchini, Menegazzo, Morigi 00)

There exists a constant c s.t. if $G \leq S_n$ is transitive, then $d(G) \leq cn/\sqrt{\log n}$.

- Kovacs and Newman: for each prime p there exists a constant c_p s.t. for all b there exists a transitive p-subgroup $P \leq S_{p^b} = S_n$

Minimal generation of transitive groups

Theorem (Cameron Solomon Turull; Neumann 89)

If $G \leq S_n$ is transitive, $n > 4$ and $(G, n) \neq (D_8 \circ D_8, 8)$ then $d(G) < n/2$.

Theorem (Lucchini, Menegazzo, Morigi 00)

There exists a constant c s.t. if $G \leq S_n$ is transitive, then $d(G) \leq cn/\sqrt{\log n}$.

- Kovacs and Newman: for each prime p there exists a constant c_p s.t. for all b there exists a transitive p-subgroup $P \leq S_{p^b} = S_n$ with $d(P) > c_p n/\sqrt{\log n}$.

Minimal generation of transitive groups

Theorem (Cameron Solomon Turull; Neumann 89)

If $G \leq S_n$ is transitive, $n > 4$ and $(G, n) \neq (D_8 \circ D_8, 8)$ then $d(G) < n/2$.

Theorem (Lucchini, Menegazzo, Morigi 00)

There exists a constant c s.t. if $G \leq S_n$ is transitive, then $d(G) \leq cn/\sqrt{\log n}$.

- Kovacs and Newman: for each prime p there exists a constant c_p s.t. for all b there exists a transitive p-subgroup $P \leq S_{p^b} = S_n$ with $d(P) > c_p n/\sqrt{\log n}$.

Theorem (Tracey 17)

Can take $c = 0.92$, or $\sqrt{3}/2$ with finitely many exceptions.

(All logs to base 2, unless otherwise stated.)

Minimal generation of transitive groups

Theorem (Cameron Solomon Turull; Neumann 89)

If $G \leq S_n$ is transitive, $n > 4$ and $(G, n) \neq (D_8 \circ D_8, 8)$ then $d(G) < n/2$.

Theorem (Lucchini, Menegazzo, Morigi 00)

There exists a constant c s.t. if $G \leq S_n$ is transitive, then $d(G) \leq cn/\sqrt{\log n}$.

- Kovacs and Newman: for each prime p there exists a constant c_p s.t. for all b there exists a transitive p-subgroup $P \leq S_{p^b} = S_n$ with $d(P) > c_p n/\sqrt{\log n}$.

Theorem (Tracey 17)

Can take $c = 0.92$, or $\sqrt{3}/2$ with finitely many exceptions.

(All logs to base 2, unless otherwise stated.)

Minimal generation of primitive groups

Let $\Delta \subseteq \{1, \ldots, n\}$.

Minimal generation of primitive groups

Let $\Delta \subseteq \{1, \ldots, n\}$. If for all $g \in G$, either $\Delta^g = \Delta$

Minimal generation of primitive groups

Let $\Delta \subseteq \{1, \ldots, n\}$. If for all $g \in G$, either $\Delta^g = \Delta$ or $\Delta^g \cap \Delta = \emptyset$

Minimal generation of primitive groups

Let $\Delta \subseteq \{1, \ldots, n\}$. If for all $g \in G$, either $\Delta^g = \Delta$ or $\Delta^g \cap \Delta = \emptyset$, then Δ is a block for G.

Minimal generation of primitive groups

Let $\Delta \subseteq \{1, \ldots, n\}$. If for all $g \in G$, either $\Delta^g = \Delta$ or $\Delta^g \cap \Delta = \emptyset$, then Δ is a block for G.

$G \leq S_n$ is primitive if G is transitive

Minimal generation of primitive groups

Let $\Delta \subseteq \{1, \ldots, n\}$. If for all $g \in G$, either $\Delta^g = \Delta$ or $\Delta^g \cap \Delta = \emptyset$, then Δ is a block for G.

$G \leq S_n$ is primitive if G is transitive and all blocks have size 1 or n.

Minimal generation of primitive groups

Let $\Delta \subseteq \{1, \ldots, n\}$. If for all $g \in G$, either $\Delta^g = \Delta$ or $\Delta^g \cap \Delta = \emptyset$, then Δ is a block for G.

$G \leq S_n$ is primitive if G is transitive and all blocks have size 1 or n.

Theorem (Lucchini, Menegazzo & Morigi 01)

There exists a constant c

Minimal generation of primitive groups

Let $\Delta \subseteq \{1, \ldots, n\}$. If for all $g \in G$, either $\Delta^g = \Delta$ or $\Delta^g \cap \Delta = \emptyset$, then Δ is a block for G.

$G \leq S_n$ is primitive if G is transitive and all blocks have size 1 or n.

Theorem (Lucchini, Menegazzo & Morigi 01)

There exists a constant c such that if $G \leq S_n$ is primitive

Minimal generation of primitive groups

Let $\Delta \subseteq \{1, \ldots, n\}$. If for all $g \in G$, either $\Delta^g = \Delta$ or $\Delta^g \cap \Delta = \emptyset$, then Δ is a block for G.

$G \leq S_n$ is primitive if G is transitive and all blocks have size 1 or n.

Theorem (Lucchini, Menegazzo & Morigi 01)

There exists a constant c such that if $G \leq S_n$ is primitive then

$$d(G) \leq \frac{c \log n}{\sqrt{\log \log n}}.$$

Minimal generation of primitive groups

Let $\Delta \subseteq \{1, \ldots, n\}$. If for all $g \in G$, either $\Delta^g = \Delta$ or $\Delta^g \cap \Delta = \emptyset$, then Δ is a block for G.
$G \leq S_n$ is primitive if G is transitive and all blocks have size 1 or n.

Theorem (Lucchini, Menegazzo & Morigi 01)

There exists a constant c such that if $G \leq S_n$ is primitive then

$$d(G) \leq \frac{c \log n}{\sqrt{\log \log n}}.$$

Theorem (Holt & CMRD 12)

Let $G \leq S_n$ be a subnormal subgroup of a primitive group.

Minimal generation of primitive groups

Let $\Delta \subseteq \{1, \ldots, n\}$. If for all $g \in G$, either $\Delta^g = \Delta$ or $\Delta^g \cap \Delta = \emptyset$, then Δ is a block for G.
$G \leq S_n$ is primitive if G is transitive and all blocks have size 1 or n.

Theorem (Lucchini, Menegazzo & Morigi 01)

There exists a constant c such that if $G \leq S_n$ is primitive then

$$d(G) \leq \frac{c \log n}{\sqrt{\log \log n}}.$$

Theorem (Holt & CMRD 12)

Let $G \leq S_n$ be a subnormal subgroup of a primitive group. Then

$$d(G) \leq \max\{\log n, 2\}.$$

Minimal generation of primitive groups

Let $\Delta \subseteq \{1, \ldots, n\}$. If for all $g \in G$, either $\Delta^g = \Delta$ or $\Delta^g \cap \Delta = \emptyset$, then Δ is a block for G.
$G \leq S_n$ is primitive if G is transitive and all blocks have size 1 or n.

Theorem (Lucchini, Menegazzo & Morigi 01)

There exists a constant c such that if $G \leq S_n$ is primitive then

$$d(G) \leq \frac{c \log n}{\sqrt{\log \log n}}.$$

Theorem (Holt & CMRD 12)

Let $G \leq S_n$ be a subnormal subgroup of a primitive group. Then

$$d(G) \leq \max\{\log n, 2\}.$$

Bound is best possible:

Minimal generation of primitive groups

Let $\Delta \subseteq \{1, \ldots, n\}$. If for all $g \in G$, either $\Delta^g = \Delta$ or $\Delta^g \cap \Delta = \emptyset$, then Δ is a block for G.
$G \leq S_n$ is primitive if G is transitive and all blocks have size 1 or n.

Theorem (Lucchini, Menegazzo & Morigi 01)

There exists a constant c such that if $G \leq S_n$ is primitive then

$$d(G) \leq \frac{c \log n}{\sqrt{\log \log n}}.$$

Theorem (Holt & CMRD 12)

Let $G \leq S_n$ be a subnormal subgroup of a primitive group. Then

$$d(G) \leq \max\{\log n, 2\}.$$

Bound is best possible: Consider $K = (\mathbb{F}_2^m, +) \trianglelefteq \mathrm{AGL}_m(2) \leq S_{2^m}$.

Minimal generation of primitive groups

Let $\Delta \subseteq \{1, \ldots, n\}$. If for all $g \in G$, either $\Delta^g = \Delta$ or $\Delta^g \cap \Delta = \emptyset$, then Δ is a block for G.
$G \leq S_n$ is primitive if G is transitive and all blocks have size 1 or n.

Theorem (Lucchini, Menegazzo & Morigi 01)

There exists a constant c such that if $G \leq S_n$ is primitive then

$$d(G) \leq \frac{c \log n}{\sqrt{\log \log n}}.$$

Theorem (Holt & CMRD 12)

Let $G \leq S_n$ be a subnormal subgroup of a primitive group. Then

$$d(G) \leq \max\{\log n, 2\}.$$

Bound is best possible: Consider $K = (\mathbb{F}_2^m, +) \trianglelefteq \mathrm{AGL}_m(2) \leq S_{2^m}$.

Random generation of permutation groups

Random generation of permutation groups

Let $d^\epsilon(G)$ be the minimum number of random elements needed to generate G with probability at least $1 - \epsilon$.

Random generation of permutation groups

Let $d^{\epsilon}(G)$ be the minimum number of random elements needed to generate G with probability at least $1 - \epsilon$.

Theorem (Various people)

Random generation of permutation groups

Let $d^{\epsilon}(G)$ be the minimum number of random elements needed to generate G with probability at least $1 - \epsilon$.

Theorem (Various people)

Let $\epsilon \in (0, 1)$

Random generation of permutation groups

Let $d^{\epsilon}(G)$ be the minimum number of random elements needed to generate G with probability at least $1 - \epsilon$.

Theorem (Various people)

Let $\epsilon \in (0, 1)$, and let t be such that $\zeta(t) \leq 1 + \epsilon$.

Random generation of permutation groups

Let $d^{\epsilon}(G)$ be the minimum number of random elements needed to generate G with probability at least $1 - \epsilon$.

Theorem (Various people)

Let $\epsilon \in (0, 1)$, and let t be such that $\zeta(t) \leq 1 + \epsilon$.
Let $G \leq S_n$.

Random generation of permutation groups

Let $d^{\epsilon}(G)$ be the minimum number of random elements needed to generate G with probability at least $1 - \epsilon$.

Theorem (Various people)

Let $\epsilon \in (0, 1)$, and let t be such that $\zeta(t) \leq 1 + \epsilon$.
Let $G \leq S_n$. Then

- $d^{\epsilon}(G) < n/2 + 2(\log n + \log \log n) + t + 2$.

Random generation of permutation groups

Let $d^\epsilon(G)$ be the minimum number of random elements needed to generate G with probability at least $1 - \epsilon$.

Theorem (Various people)

Let $\epsilon \in (0, 1)$, and let t be such that $\zeta(t) \leq 1 + \epsilon$.
Let $G \leq S_n$. Then

- $d^\epsilon(G) < n/2 + 2(\log n + \log\log n) + t + 2$.
- *If G is transitive then*
  $d^\epsilon(G) < \frac{0.92n}{\sqrt{\log n}} + 2(\log n + \log\log n) + t + 2$.

Random generation of permutation groups

Let $d^{\epsilon}(G)$ be the minimum number of random elements needed to generate G with probability at least $1 - \epsilon$.

Theorem (Various people)

Let $\epsilon \in (0, 1)$, and let t be such that $\zeta(t) \leq 1 + \epsilon$.
Let $G \leq S_n$. Then

- $d^{\epsilon}(G) < n/2 + 2(\log n + \log\log n) + t + 2$.
- *If G is transitive then* $d^{\epsilon}(G) < \frac{0.92n}{\sqrt{\log n}} + 2(\log n + \log\log n) + t + 2$.
- *If G is a subnormal subgroup of a primitive group*

Random generation of permutation groups

Let $d^\epsilon(G)$ be the minimum number of random elements needed to generate G with probability at least $1 - \epsilon$.

Theorem (Various people)

Let $\epsilon \in (0, 1)$, and let t be such that $\zeta(t) \leq 1 + \epsilon$.
Let $G \leq S_n$. Then

- $d^\epsilon(G) < n/2 + 2(\log n + \log\log n) + t + 2$.
- *If G is transitive then* $d^\epsilon(G) < \frac{0.92n}{\sqrt{\log n}} + 2(\log n + \log\log n) + t + 2$.
- *If G is a subnormal subgroup of a primitive group, then* $d^\epsilon(G) < 3\log n + 2\log\log n + t + 2$.

Random generation of permutation groups

Let $d^\epsilon(G)$ be the minimum number of random elements needed to generate G with probability at least $1 - \epsilon$.

Theorem (Various people)

Let $\epsilon \in (0, 1)$, and let t be such that $\zeta(t) \leq 1 + \epsilon$.
Let $G \leq S_n$. Then

- $d^\epsilon(G) < n/2 + 2(\log n + \log\log n) + t + 2$.
- *If G is transitive then* $d^\epsilon(G) < \frac{0.92n}{\sqrt{\log n}} + 2(\log n + \log\log n) + t + 2$.
- *If G is a subnormal subgroup of a primitive group, then* $d^\epsilon(G) < 3\log n + 2\log\log n + t + 2$.
- *If G is primitive then* $d^\epsilon(G) < \log n + \log\log n + t + 4.59$.

Random generation of permutation groups

Let $d^\epsilon(G)$ be the minimum number of random elements needed to generate G with probability at least $1 - \epsilon$.

Theorem (Various people)

Let $\epsilon \in (0, 1)$, and let t be such that $\zeta(t) \leq 1 + \epsilon$.
Let $G \leq S_n$. Then

- $d^\epsilon(G) < n/2 + 2(\log n + \log\log n) + t + 2$.
- *If G is transitive then* $d^\epsilon(G) < \frac{0.92n}{\sqrt{\log n}} + 2(\log n + \log\log n) + t + 2$.
- *If G is a subnormal subgroup of a primitive group, then* $d^\epsilon(G) < 3\log n + 2\log\log n + t + 2$.
- *If G is primitive then* $d^\epsilon(G) < \log n + \log\log n + t + 4.59$.

What is a random subgroup of S_n?

What is a random subgroup of S_n?

More than one possible interpretation of "random subgroup":

What is a random subgroup of S_n?

More than one possible interpretation of "random subgroup":

- Uniformly randomly generated?

What is a random subgroup of S_n?

More than one possible interpretation of "random subgroup":

- Uniformly randomly generated? With probability tending rapidly to 1, this is S_n

What is a random subgroup of S_n?

More than one possible interpretation of "random subgroup":

- Uniformly randomly generated? With probability tending rapidly to 1, this is S_n (or cyclic, or A_n).

What is a random subgroup of S_n?

More than one possible interpretation of "random subgroup":

- Uniformly randomly generated? With probability tending rapidly to 1, this is S_n (or cyclic, or A_n).
- Uniform random amongst the subgroups of S_n?

What is a random subgroup of S_n?

More than one possible interpretation of "random subgroup":

- Uniformly randomly generated? With probability tending rapidly to 1, this is S_n (or cyclic, or A_n).
- Uniform random amongst the subgroups of S_n?
- Uniform random amongst the conjugacy classes of subgroups?

What is a random subgroup of S_n?

More than one possible interpretation of "random subgroup":

- Uniformly randomly generated? With probability tending rapidly to 1, this is S_n (or cyclic, or A_n).
- Uniform random amongst the subgroups of S_n?
- Uniform random amongst the conjugacy classes of subgroups?

Theorem (Pyber 93)

$a(n)$ – number of subgroups of S_n.

What is a random subgroup of S_n?

More than one possible interpretation of "random subgroup":

- Uniformly randomly generated? With probability tending rapidly to 1, this is S_n (or cyclic, or A_n).
- Uniform random amongst the subgroups of S_n?
- Uniform random amongst the conjugacy classes of subgroups?

Theorem (Pyber 93)

$a(n)$ – number of subgroups of S_n. Then

$$2^{n^2(1/16+o(1))} \leq a(n) \leq 2^{n^2(\log_2(24)/6+o(1))}.$$

What is a random subgroup of S_n?

More than one possible interpretation of "random subgroup":

- Uniformly randomly generated? With probability tending rapidly to 1, this is S_n (or cyclic, or A_n).
- Uniform random amongst the subgroups of S_n?
- Uniform random amongst the conjugacy classes of subgroups?

Theorem (Pyber 93)

$a(n)$ – number of subgroups of S_n. Then

$$2^{n^2(1/16+o(1))} \leq a(n) \leq 2^{n^2(\log_2(24)/6+o(1))}.$$

Lower bound:

What is a random subgroup of S_n?

More than one possible interpretation of "random subgroup":

- Uniformly randomly generated? With probability tending rapidly to 1, this is S_n (or cyclic, or A_n).
- Uniform random amongst the subgroups of S_n?
- Uniform random amongst the conjugacy classes of subgroups?

Theorem (Pyber 93)

$a(n)$ – number of subgroups of S_n. Then

$$2^{n^2(1/16+o(1))} \leq a(n) \leq 2^{n^2(\log_2(24)/6+o(1))}.$$

Lower bound: consider $C_2^{\lfloor n/2 \rfloor} \cong \mathbb{F}_2^{\lfloor n/2 \rfloor} < S_n$

What is a random subgroup of S_n?

More than one possible interpretation of "random subgroup":

- Uniformly randomly generated? With probability tending rapidly to 1, this is S_n (or cyclic, or A_n).
- Uniform random amongst the subgroups of S_n?
- Uniform random amongst the conjugacy classes of subgroups?

Theorem (Pyber 93)

$a(n)$ – number of subgroups of S_n. Then

$$2^{n^2(1/16+o(1))} \leq a(n) \leq 2^{n^2(\log_2(24)/6+o(1))}.$$

Lower bound: consider $C_2^{\lfloor n/2 \rfloor} \cong \mathbb{F}_2^{\lfloor n/2 \rfloor} < S_n$, and count subspaces.

What is a random subgroup of S_n?

More than one possible interpretation of "random subgroup":

- Uniformly randomly generated? With probability tending rapidly to 1, this is S_n (or cyclic, or A_n).
- Uniform random amongst the subgroups of S_n?
- Uniform random amongst the conjugacy classes of subgroups?

Theorem (Pyber 93)

$a(n)$ – number of subgroups of S_n. Then

$$2^{n^2(1/16+o(1))} \leq a(n) \leq 2^{n^2(\log_2(24)/6+o(1))}.$$

Lower bound: consider $C_2^{\lfloor n/2 \rfloor} \cong \mathbb{F}_2^{\lfloor n/2 \rfloor} < S_n$, and count subspaces.

Hence: not much difference between "random amongst subgroups" and "random amongst conjugacy classes of subgroups".

What is a random subgroup of S_n?

More than one possible interpretation of "random subgroup":

- Uniformly randomly generated? With probability tending rapidly to 1, this is S_n (or cyclic, or A_n).
- Uniform random amongst the subgroups of S_n?
- Uniform random amongst the conjugacy classes of subgroups?

Theorem (Pyber 93)

$a(n)$ – number of subgroups of S_n. Then

$$2^{n^2(1/16+o(1))} \leq a(n) \leq 2^{n^2(\log_2(24)/6+o(1))}.$$

Lower bound: consider $C_2^{\lfloor n/2 \rfloor} \cong \mathbb{F}_2^{\lfloor n/2 \rfloor} < S_n$, and count subspaces.

Hence: not much difference between "random amongst subgroups" and "random amongst conjugacy classes of subgroups".

More on random subgroups of S_n

More on random subgroups of S_n

$\mathcal{P}$ – property of permutation groups.

More on random subgroups of S_n

$\mathcal{P}$ – property of permutation groups. If have a bound $f_{\mathcal{P}}(n)$ on the number of generators of a subgroup of S_n with property $\mathcal{P}$

More on random subgroups of S_n

$\mathcal{P}$ – property of permutation groups. If have a bound $f_{\mathcal{P}}(n)$ on the number of generators of a subgroup of S_n with property $\mathcal{P}$, then there are at most $(n!)^{f_{\mathcal{P}}(n)} < 2^{f_{\mathcal{P}}(n) n \log n}$ subgroups with $\mathcal{P}$.

More on random subgroups of S_n

$\mathcal{P}$ – property of permutation groups. If have a bound $f_{\mathcal{P}}(n)$ on the number of generators of a subgroup of S_n with property $\mathcal{P}$, then there are at most $(n!)^{f_{\mathcal{P}}(n)} < 2^{f_{\mathcal{P}}(n) n \log n}$ subgroups with $\mathcal{P}$.

Corollary

$\mathcal{P}$ – property such that $f_{\mathcal{P}}(n) < \frac{n}{(\log n)^{1+\varepsilon}}$ for $\varepsilon > 0$.

More on random subgroups of S_n

$\mathcal{P}$ – property of permutation groups. If have a bound $f_{\mathcal{P}}(n)$ on the number of generators of a subgroup of S_n with property $\mathcal{P}$, then there are at most $(n!)^{f_{\mathcal{P}}(n)} < 2^{f_{\mathcal{P}}(n) n \log n}$ subgroups with $\mathcal{P}$.

Corollary

$\mathcal{P}$ – property such that $f_{\mathcal{P}}(n) < \frac{n}{(\log n)^{1+\varepsilon}}$ for $\varepsilon > 0$. Then the proportion of subgroups of S_n that satisfy $\mathcal{P}$

More on random subgroups of S_n

$\mathcal{P}$ – property of permutation groups. If have a bound $f_{\mathcal{P}}(n)$ on the number of generators of a subgroup of S_n with property $\mathcal{P}$, then there are at most $(n!)^{f_{\mathcal{P}}(n)} < 2^{f_{\mathcal{P}}(n) n \log n}$ subgroups with $\mathcal{P}$.

Corollary

$\mathcal{P}$ – property such that $f_{\mathcal{P}}(n) < \frac{n}{(\log n)^{1+\varepsilon}}$ for $\varepsilon > 0$. Then the proportion of subgroups of S_n that satisfy $\mathcal{P}$ tends to 0 as $n \to \infty$.

More on random subgroups of S_n

$\mathcal{P}$ – property of permutation groups. If have a bound $f_{\mathcal{P}}(n)$ on the number of generators of a subgroup of S_n with property $\mathcal{P}$, then there are at most $(n!)^{f_{\mathcal{P}}(n)} < 2^{f_{\mathcal{P}}(n) n \log n}$ subgroups with $\mathcal{P}$.

Corollary

$\mathcal{P}$ – property such that $f_{\mathcal{P}}(n) < \frac{n}{(\log n)^{1+\varepsilon}}$ for $\varepsilon > 0$. Then the proportion of subgroups of S_n that satisfy $\mathcal{P}$ tends to 0 as $n \to \infty$.

Theorem (Lucchini, Menegazzo, Morigi 00)

There exists a constant b such that the number of transitive subgroups of S_n

More on random subgroups of S_n

$\mathcal{P}$ – property of permutation groups. If have a bound $f_{\mathcal{P}}(n)$ on the number of generators of a subgroup of S_n with property $\mathcal{P}$, then there are at most $(n!)^{f_{\mathcal{P}}(n)} < 2^{f_{\mathcal{P}}(n) n \log n}$ subgroups with $\mathcal{P}$.

Corollary

$\mathcal{P}$ – property such that $f_{\mathcal{P}}(n) < \frac{n}{(\log n)^{1+\varepsilon}}$ for $\varepsilon > 0$. Then the proportion of subgroups of S_n that satisfy $\mathcal{P}$ tends to 0 as $n \to \infty$.

Theorem (Lucchini, Menegazzo, Morigi 00)

There exists a constant b such that the number of transitive subgroups of S_n is at most

$$2^{bn^2/\sqrt{\log n}}.$$

More on random subgroups of S_n

$\mathcal{P}$ – property of permutation groups. If have a bound $f_{\mathcal{P}}(n)$ on the number of generators of a subgroup of S_n with property $\mathcal{P}$, then there are at most $(n!)^{f_{\mathcal{P}}(n)} < 2^{f_{\mathcal{P}}(n) n \log n}$ subgroups with $\mathcal{P}$.

Corollary

$\mathcal{P}$ – property such that $f_{\mathcal{P}}(n) < \frac{n}{(\log n)^{1+\varepsilon}}$ for $\varepsilon > 0$. Then the proportion of subgroups of S_n that satisfy $\mathcal{P}$ tends to 0 as $n \to \infty$.

Theorem (Lucchini, Menegazzo, Morigi 00)

There exists a constant b such that the number of transitive subgroups of S_n is at most

$$2^{bn^2/\sqrt{\log n}}.$$

Hence the proportion of subgroups of S_n that are transitive tends to 0 as $n \to \infty$.

More on random subgroups of S_n

$\mathcal{P}$ – property of permutation groups. If have a bound $f_{\mathcal{P}}(n)$ on the number of generators of a subgroup of S_n with property $\mathcal{P}$, then there are at most $(n!)^{f_{\mathcal{P}}(n)} < 2^{f_{\mathcal{P}}(n) n \log n}$ subgroups with $\mathcal{P}$.

Corollary

$\mathcal{P}$ – property such that $f_{\mathcal{P}}(n) < \frac{n}{(\log n)^{1+\varepsilon}}$ for $\varepsilon > 0$. Then the proportion of subgroups of S_n that satisfy $\mathcal{P}$ tends to 0 as $n \to \infty$.

Theorem (Lucchini, Menegazzo, Morigi 00)

There exists a constant b such that the number of transitive subgroups of S_n is at most

$$2^{bn^2/\sqrt{\log n}}.$$

Hence the proportion of subgroups of S_n that are transitive tends to 0 as $n \to \infty$.

Some speculation

Some speculation

It looks likely that a random subgroup of S_n should be

Some speculation

It looks likely that a random subgroup of S_n should be

- "Close" to soluble.

Some speculation

It looks likely that a random subgroup of S_n should be

- "Close" to soluble.
- Have all orbits "short".

Some speculation

It looks likely that a random subgroup of S_n should be

- "Close" to soluble.
- Have all orbits "short".
- Have order dominated by that of its Sylow 2-subgroup.

Some speculation

It looks likely that a random subgroup of S_n should be

- "Close" to soluble.
- Have all orbits "short".
- Have order dominated by that of its Sylow 2-subgroup.

Computational group theory

Computational group theory

General set-up when computing with a finite permutation group:

Computational group theory

General set-up when computing with a finite permutation group:

- Input:

Computational group theory

General set-up when computing with a finite permutation group:

- Input: A set $X \subset S_n$ of generators for a group G.

Computational group theory

General set-up when computing with a finite permutation group:

- Input: A set $X \subset S_n$ of generators for a group G.
- Output: answers to questions about G.

Computational group theory

General set-up when computing with a finite permutation group:

- Input: A set $X \subset S_n$ of generators for a group G.
- Output: answers to questions about G.

Input size is $|X| n \log n$

Computational group theory

General set-up when computing with a finite permutation group:

- Input: A set $X \subset S_n$ of generators for a group G.
- Output: answers to questions about G.

Input size is $|X|n \log n$, so the complexity of any given algorithm

Computational group theory

General set-up when computing with a finite permutation group:

- Input: A set $X \subset S_n$ of generators for a group G.
- Output: answers to questions about G.

Input size is $|X|n \log n$, so the complexity of any given algorithm is a function of $|X|$ and n.

Computational group theory

General set-up when computing with a finite permutation group:

- Input: A set $X \subset S_n$ of generators for a group G.
- Output: answers to questions about G.

Input size is $|X| n \log n$, so the complexity of any given algorithm is a function of $|X|$ and n.
There are effective methods to reduce X to a "useful" set of size $O(n)$, so complexity is a normally a function of n.

Traditionally, looked at worst case complexity.

Computational group theory

General set-up when computing with a finite permutation group:

- Input: A set $X \subset S_n$ of generators for a group G.
- Output: answers to questions about G.

Input size is $|X| n \log n$, so the complexity of any given algorithm is a function of $|X|$ and n.
There are effective methods to reduce X to a "useful" set of size $O(n)$, so complexity is a normally a function of n.

Traditionally, looked at worst case complexity.

Is interesting (but harder) to look at generic case complexity.

Computational group theory

General set-up when computing with a finite permutation group:

- Input: A set $X \subset S_n$ of generators for a group G.
- Output: answers to questions about G.

Input size is $|X| n \log n$, so the complexity of any given algorithm is a function of $|X|$ and n.
There are effective methods to reduce X to a "useful" set of size $O(n)$, so complexity is a normally a function of n.

Traditionally, looked at worst case complexity.

Is interesting (but harder) to look at generic case complexity.

- Consider a subset A_n of the set B_n of all possible inputs, such that $|A_n|/|B_n| \to 1$ as $n \to \infty$.

Computational group theory

General set-up when computing with a finite permutation group:

- Input: A set $X \subset S_n$ of generators for a group G.
- Output: answers to questions about G.

Input size is $|X| n \log n$, so the complexity of any given algorithm is a function of $|X|$ and n.
There are effective methods to reduce X to a "useful" set of size $O(n)$, so complexity is a normally a function of n.

Traditionally, looked at worst case complexity.

Is interesting (but harder) to look at generic case complexity.

- Consider a subset A_n of the set B_n of all possible inputs, such that $|A_n|/|B_n| \to 1$ as $n \to \infty$.
- Measure the worst-case complexity on this set.

Computational group theory

General set-up when computing with a finite permutation group:

- Input: A set $X \subset S_n$ of generators for a group G.
- Output: answers to questions about G.

Input size is $|X| n \log n$, so the complexity of any given algorithm is a function of $|X|$ and n.
There are effective methods to reduce X to a "useful" set of size $O(n)$, so complexity is a normally a function of n.

Traditionally, looked at worst case complexity.

Is interesting (but harder) to look at generic case complexity.

- Consider a subset A_n of the set B_n of all possible inputs, such that $|A_n|/|B_n| \to 1$ as $n \to \infty$.
- Measure the worst-case complexity on this set.

That is, we're allowed to ignore some groups $G \leq S_n$, for each n.

Computational group theory

General set-up when computing with a finite permutation group:

- Input: A set $X \subset S_n$ of generators for a group G.
- Output: answers to questions about G.

Input size is $|X| n \log n$, so the complexity of any given algorithm is a function of $|X|$ and n.
There are effective methods to reduce X to a "useful" set of size $O(n)$, so complexity is a normally a function of n.

Traditionally, looked at worst case complexity.

Is interesting (but harder) to look at generic case complexity.

- Consider a subset A_n of the set B_n of all possible inputs, such that $|A_n|/|B_n| \to 1$ as $n \to \infty$.
- Measure the worst-case complexity on this set.

That is, we're allowed to ignore some groups $G \leq S_n$, for each n.

Classes P and NP

Classes P and NP

The class P consists of the problems that can be solved in time a polynomial in their input size.

Classes P and NP

The class P consists of the problems that can be solved in time a polynomial in their input size.

- For permutation groups: polynomial in n.

Classes P and NP

The class P consists of the problems that can be solved in time a polynomial in their input size.

- For permutation groups: polynomial in n.

The class NP consists of the problems whose solution can be verified in time a polynomial in their input size.

Classes P and NP

The class P consists of the problems that can be solved in time a polynomial in their input size.

- For permutation groups: polynomial in n.

The class NP consists of the problems whose solution can be verified in time a polynomial in their input size.

- Strictly speaking these are decision problems

Classes P and NP

The class P consists of the problems that can be solved in time a polynomial in their input size.

- For permutation groups: polynomial in n.

The class NP consists of the problems whose solution can be verified in time a polynomial in their input size.

- Strictly speaking these are decision problems, but often they are equivalent to problems with other types of answers.

Classes P and NP

The class P consists of the problems that can be solved in time a polynomial in their input size.

- For permutation groups: polynomial in n.

The class NP consists of the problems whose solution can be verified in time a polynomial in their input size.

- Strictly speaking these are decision problems, but often they are equivalent to problems with other types of answers.

Some problems known to be in NP but not known to be in P:

Classes P and NP

The class P consists of the problems that can be solved in time a polynomial in their input size.

- For permutation groups: polynomial in n.

The class NP consists of the problems whose solution can be verified in time a polynomial in their input size.

- Strictly speaking these are decision problems, but often they are equivalent to problems with other types of answers.

Some problems known to be in NP but not known to be in P:

- Graph isomorphism:

Classes P and NP

The class P consists of the problems that can be solved in time a polynomial in their input size.

- For permutation groups: polynomial in n.

The class NP consists of the problems whose solution can be verified in time a polynomial in their input size.

- Strictly speaking these are decision problems, but often they are equivalent to problems with other types of answers.

Some problems known to be in NP but not known to be in P:

- Graph isomorphism: Given two graphs Γ_1, Γ_2

Classes P and NP

The class P consists of the problems that can be solved in time a polynomial in their input size.

- For permutation groups: polynomial in n.

The class NP consists of the problems whose solution can be verified in time a polynomial in their input size.

- Strictly speaking these are decision problems, but often they are equivalent to problems with other types of answers.

Some problems known to be in NP but not known to be in P:

- Graph isomorphism: Given two graphs Γ_1, Γ_2, decide if $\Gamma_1 \cong \Gamma_2$.

Classes P and NP

The class P consists of the problems that can be solved in time a polynomial in their input size.

- For permutation groups: polynomial in n.

The class NP consists of the problems whose solution can be verified in time a polynomial in their input size.

- Strictly speaking these are decision problems, but often they are equivalent to problems with other types of answers.

Some problems known to be in NP but not known to be in P:

- Graph isomorphism: Given two graphs Γ_1, Γ_2, decide if $\Gamma_1 \cong \Gamma_2$.
- Subgroup intersection:

Classes P and NP

The class P consists of the problems that can be solved in time a polynomial in their input size.

- For permutation groups: polynomial in n.

The class NP consists of the problems whose solution can be verified in time a polynomial in their input size.

- Strictly speaking these are decision problems, but often they are equivalent to problems with other types of answers.

Some problems known to be in NP but not known to be in P:

- Graph isomorphism: Given two graphs Γ_1, Γ_2, decide if $\Gamma_1 \cong \Gamma_2$.
- Subgroup intersection: Given $G, H \leq S_n$, find $G \cap H$.

Classes P and NP

The class P consists of the problems that can be solved in time a polynomial in their input size.

- For permutation groups: polynomial in n.

The class NP consists of the problems whose solution can be verified in time a polynomial in their input size.

- Strictly speaking these are decision problems, but often they are equivalent to problems with other types of answers.

Some problems known to be in NP but not known to be in P:

- Graph isomorphism: Given two graphs Γ_1, Γ_2, decide if $\Gamma_1 \cong \Gamma_2$.
- Subgroup intersection: Given $G, H \leq S_n$, find $G \cap H$.
- Set stabiliser: Given $G \leq S_n$, $\Delta \subseteq \{1, \dots n\}$, find $G_{\{\Delta\}}$.
- Normaliser:

Classes P and NP

The class P consists of the problems that can be solved in time a polynomial in their input size.

- For permutation groups: polynomial in n.

The class NP consists of the problems whose solution can be verified in time a polynomial in their input size.

- Strictly speaking these are decision problems, but often they are equivalent to problems with other types of answers.

Some problems known to be in NP but not known to be in P:

- Graph isomorphism: Given two graphs Γ_1, Γ_2, decide if $\Gamma_1 \cong \Gamma_2$.
- Subgroup intersection: Given $G, H \leq S_n$, find $G \cap H$.
- Set stabiliser: Given $G \leq S_n$, $\Delta \subseteq \{1, \dots n\}$, find $G_{\{\Delta\}}$.
- Normaliser: Given $G, H \leq S_n$,

Classes P and NP

The class P consists of the problems that can be solved in time a polynomial in their input size.

- For permutation groups: polynomial in n.

The class NP consists of the problems whose solution can be verified in time a polynomial in their input size.

- Strictly speaking these are decision problems, but often they are equivalent to problems with other types of answers.

Some problems known to be in NP but not known to be in P:

- Graph isomorphism: Given two graphs Γ_1, Γ_2, decide if $\Gamma_1 \cong \Gamma_2$.
- Subgroup intersection: Given $G, H \leq S_n$, find $G \cap H$.
- Set stabiliser: Given $G \leq S_n$, $\Delta \subseteq \{1, \ldots n\}$, find $G_{\{\Delta\}}$.
- Normaliser: Given $G, H \leq S_n$, find $N_G(H)$.

Classes P and NP

The class P consists of the problems that can be solved in time a polynomial in their input size.

- For permutation groups: polynomial in n.

The class NP consists of the problems whose solution can be verified in time a polynomial in their input size.

- Strictly speaking these are decision problems, but often they are equivalent to problems with other types of answers.

Some problems known to be in NP but not known to be in P:

- Graph isomorphism: Given two graphs Γ_1, Γ_2, decide if $\Gamma_1 \cong \Gamma_2$.
- Subgroup intersection: Given $G, H \leq S_n$, find $G \cap H$.
- Set stabiliser: Given $G \leq S_n$, $\Delta \subseteq \{1, \ldots n\}$, find $G_{\{\Delta\}}$.
- Normaliser: Given $G, H \leq S_n$, find $N_G(H)$.

Progress on hard permutation group problems

Progress on hard permutation group problems

A problem $\mathcal{P}$ is polynomial-time reducible to a problem $\mathcal{Q}$ if a polynomial-time soln to $\mathcal{Q}$ would yield a polynomial-time soln to $\mathcal{P}$.

Theorem (Luks 93)

- *Graph isomorphism is polynomial-time reducible to subgroup intersection.*

Progress on hard permutation group problems

A problem $\mathcal{P}$ is polynomial-time reducible to a problem $\mathcal{Q}$ if a polynomial-time soln to $\mathcal{Q}$ would yield a polynomial-time soln to $\mathcal{P}$.

Theorem (Luks 93)

- *Graph isomorphism is polynomial-time reducible to subgroup intersection.*
- *Subgroup intersection and set stabiliser are polynomial-time equivalent.*

Progress on hard permutation group problems

A problem $\mathcal{P}$ is polynomial-time reducible to a problem $\mathcal{Q}$ if a polynomial-time soln to $\mathcal{Q}$ would yield a polynomial-time soln to $\mathcal{P}$.

Theorem (Luks 93)

- *Graph isomorphism is polynomial-time reducible to subgroup intersection.*
- *Subgroup intersection and set stabiliser are polynomial-time equivalent.*
- *Subgroup intersection is polynomial-time reducible to normaliser.*

Progress on hard permutation group problems

A problem $\mathcal{P}$ is polynomial-time reducible to a problem $\mathcal{Q}$ if a polynomial-time soln to $\mathcal{Q}$ would yield a polynomial-time soln to $\mathcal{P}$.

Theorem (Luks 93)

- *Graph isomorphism is polynomial-time reducible to subgroup intersection.*
- *Subgroup intersection and set stabiliser are polynomial-time equivalent.*
- *Subgroup intersection is polynomial-time reducible to normaliser.*

Progress on hard permutation group problems

A problem $\mathcal{P}$ is polynomial-time reducible to a problem $\mathcal{Q}$ if a polynomial-time soln to $\mathcal{Q}$ would yield a polynomial-time soln to $\mathcal{P}$.

Theorem (Luks 93)

- *Graph isomorphism is polynomial-time reducible to subgroup intersection.*
- *Subgroup intersection and set stabiliser are polynomial-time equivalent.*
- *Subgroup intersection is polynomial-time reducible to normaliser.*

Theorem (Babai 2017)

The graph isomorphism problem for a pair of graphs on n vertices has complexity $O(2^{(\log n)^c})$

Progress on hard permutation group problems

A problem $\mathcal{P}$ is polynomial-time reducible to a problem $\mathcal{Q}$ if a polynomial-time soln to $\mathcal{Q}$ would yield a polynomial-time soln to $\mathcal{P}$.

Theorem (Luks 93)

- *Graph isomorphism is polynomial-time reducible to subgroup intersection.*
- *Subgroup intersection and set stabiliser are polynomial-time equivalent.*
- *Subgroup intersection is polynomial-time reducible to normaliser.*

Theorem (Babai 2017)

The graph isomorphism problem for a pair of graphs on n vertices has complexity $O(2^{(\log n)^c})$, for a fixed constant c.

Progress on hard permutation group problems

A problem $\mathcal{P}$ is polynomial-time reducible to a problem $\mathcal{Q}$ if a polynomial-time soln to $\mathcal{Q}$ would yield a polynomial-time soln to $\mathcal{P}$.

Theorem (Luks 93)

- *Graph isomorphism is polynomial-time reducible to subgroup intersection.*
- *Subgroup intersection and set stabiliser are polynomial-time equivalent.*
- *Subgroup intersection is polynomial-time reducible to normaliser.*

Theorem (Babai 2017)

The graph isomorphism problem for a pair of graphs on n vertices has complexity $O(2^{(\log n)^c})$, for a fixed constant c.

Helfgott 2017: Can take $c = 3$.

Progress on hard permutation group problems

A problem $\mathcal{P}$ is polynomial-time reducible to a problem $\mathcal{Q}$ if a polynomial-time soln to $\mathcal{Q}$ would yield a polynomial-time soln to $\mathcal{P}$.

Theorem (Luks 93)

- *Graph isomorphism is polynomial-time reducible to subgroup intersection.*
- *Subgroup intersection and set stabiliser are polynomial-time equivalent.*
- *Subgroup intersection is polynomial-time reducible to normaliser.*

Theorem (Babai 2017)

The graph isomorphism problem for a pair of graphs on n vertices has complexity $O(2^{(\log n)^c})$, for a fixed constant c.

Helfgott 2017: Can take $c = 3$.

Polynomial-time results for special cases

Polynomial-time results for special cases

Let $\Gamma_d = \{H \leq S_n \ : \text{every nonabelian composition factor of } H \text{ is isomorphic to a subgroup of } \mathrm{S}_d\}$.

Polynomial-time results for special cases

Let $\Gamma_d = \{H \leq S_n \ : \text{every nonabelian composition factor of } H \text{ is isomorphic to a subgroup of } S_d\}$.
All soluble groups lie in Γ_d, for all d.

Polynomial-time results for special cases

Let $\Gamma_d = \{H \leq S_n \ : \text{every nonabelian composition factor of } H \text{ is isomorphic to a subgroup of } S_d\}$.
All soluble groups lie in Γ_d, for all d.

Theorem (Luks 93)

Fix d.

Polynomial-time results for special cases

Let $\Gamma_d = \{H \leq S_n \ : \text{every nonabelian composition factor of } H \text{ is isomorphic to a subgroup of } S_d\}$.
All soluble groups lie in Γ_d, for all d.

Theorem (Luks 93)

Fix d. Given $G \leq S_n$, with $G \in \Gamma_d$,

Polynomial-time results for special cases

Let $\Gamma_d = \{H \leq S_n \ : \text{every nonabelian composition factor of } H \text{ is isomorphic to a subgroup of } S_d\}$.
All soluble groups lie in Γ_d, for all d.

Theorem (Luks 93)

Fix d. Given $G \leq S_n$, with $G \in \Gamma_d$, in polynomial-time one can:

Polynomial-time results for special cases

Let $\Gamma_d = \{H \leq S_n \ : \text{every nonabelian composition factor of } H \text{ is isomorphic to a subgroup of } S_d\}$.
All soluble groups lie in Γ_d, for all d.

Theorem (Luks 93)

Fix d. Given $G \leq S_n$, with $G \in \Gamma_d$, in polynomial-time one can:

- *For any $\Delta \subset \{1, \ldots, n\}$, find $G_{\{\Delta\}}$.*

Polynomial-time results for special cases

Let $\Gamma_d = \{H \leq S_n \ : \text{every nonabelian composition factor of } H \text{ is isomorphic to a subgroup of } S_d\}$.
All soluble groups lie in Γ_d, for all d.

Theorem (Luks 93)

Fix d. Given $G \leq S_n$, with $G \in \Gamma_d$, in polynomial-time one can:

- *For any $\Delta \subset \{1, \ldots, n\}$, find $G_{\{\Delta\}}$.*
- *For any $H \leq S_n$, find $G \cap H$.*

Polynomial-time results for special cases

Let $\Gamma_d = \{H \leq S_n \ : \text{every nonabelian composition factor of } H \text{ is isomorphic to a subgroup of } S_d\}$.
All soluble groups lie in Γ_d, for all d.

Theorem (Luks 93)

Fix d. Given $G \leq S_n$, with $G \in \Gamma_d$, in polynomial-time one can:

- *For any $\Delta \subset \{1, \ldots, n\}$, find $G_{\{\Delta\}}$.*
- *For any $H \leq S_n$, find $G \cap H$.*

Corollary

If there exists a d s.t. a generic subgroup G of S_n lies in Γ_d, then the set stabiliser and intersection problem are generically polynomial-time.

More on the normaliser problem

More on the normaliser problem

Theorem (Luks & Miyazaki 11)

Let d be fixed.

More on the normaliser problem

Theorem (Luks & Miyazaki 11)

Let d be fixed. Given $G, H \leq S_n$, with $G \in \Gamma_d$,

More on the normaliser problem

Theorem (Luks & Miyazaki 11)

Let d be fixed. Given $G, H \leq S_n$, with $G \in \Gamma_d$, in polynomial-time one can find $N_G(H)$.

More on the normaliser problem

Theorem (Luks & Miyazaki 11)

Let d be fixed. Given $G, H \leq S_n$, with $G \in \Gamma_d$, in polynomial-time one can find $N_G(H)$.

- It is not known whether $G = S_n$ is as hard as G arbitrary.

More on the normaliser problem

Theorem (Luks & Miyazaki 11)

Let d be fixed. Given $G, H \leq S_n$, with $G \in \Gamma_d$, in polynomial-time one can find $N_G(H)$.

- It is not known whether $G = S_n$ is as hard as G arbitrary.
- Current methods to find $N := N_G(H)$ search through G.

More on the normaliser problem

Theorem (Luks & Miyazaki 11)

Let d be fixed. Given $G, H \leq S_n$, with $G \in \Gamma_d$, in polynomial-time one can find $N_G(H)$.

- It is not known whether $G = S_n$ is as hard as G arbitrary.
- Current methods to find $N := N_G(H)$ search through G.

Main methods to reduce the number of elements of G to consider:

More on the normaliser problem

Theorem (Luks & Miyazaki 11)

Let d be fixed. Given $G, H \leq S_n$, with $G \in \Gamma_d$, in polynomial-time one can find $N_G(H)$.

- It is not known whether $G = S_n$ is as hard as G arbitrary.
- Current methods to find $N := N_G(H)$ search through G.

Main methods to reduce the number of elements of G to consider:

- N permutes the orbits of H.

More on the normaliser problem

Theorem (Luks & Miyazaki 11)

Let d be fixed. Given $G, H \leq S_n$, with $G \in \Gamma_d$, in polynomial-time one can find $N_G(H)$.

- It is not known whether $G = S_n$ is as hard as G arbitrary.
- Current methods to find $N := N_G(H)$ search through G.

Main methods to reduce the number of elements of G to consider:

- N permutes the orbits of H.
- N permutes the orbital graphs of H.

More on the normaliser problem

Theorem (Luks & Miyazaki 11)

Let d be fixed. Given $G, H \leq S_n$, with $G \in \Gamma_d$, in polynomial-time one can find $N_G(H)$.

- It is not known whether $G = S_n$ is as hard as G arbitrary.
- Current methods to find $N := N_G(H)$ search through G.

Main methods to reduce the number of elements of G to consider:

- N permutes the orbits of H.
- N permutes the orbital graphs of H.
- If $g \in N$ then $(H_{(\alpha_1,\ldots,\alpha_k)})^g = H_{(\alpha_1^g,\alpha_2^g,\ldots,\alpha_k^g)}$, for all $\alpha_i \in \{1,\ldots,n\}$ and $1 \leq k \leq n$.

More on the normaliser problem

Theorem (Luks & Miyazaki 11)

Let d be fixed. Given $G, H \leq S_n$, with $G \in \Gamma_d$, in polynomial-time one can find $N_G(H)$.

- It is not known whether $G = S_n$ is as hard as G arbitrary.
- Current methods to find $N := N_G(H)$ search through G.

Main methods to reduce the number of elements of G to consider:

- N permutes the orbits of H.
- N permutes the orbital graphs of H.
- If $g \in N$ then $(H_{(\alpha_1,\ldots,\alpha_k)})^g = H_{(\alpha_1^g,\alpha_2^g,\ldots,\alpha_k^g)}$, for all $\alpha_i \in \{1,\ldots,n\}$ and $1 \leq k \leq n$.
- Testing conjugacy of subgroups is hard in general

More on the normaliser problem

Theorem (Luks & Miyazaki 11)

Let d be fixed. Given $G, H \leq S_n$, with $G \in \Gamma_d$, in polynomial-time one can find $N_G(H)$.

- It is not known whether $G = S_n$ is as hard as G arbitrary.
- Current methods to find $N := N_G(H)$ search through G.

Main methods to reduce the number of elements of G to consider:

- N permutes the orbits of H.
- N permutes the orbital graphs of H.
- If $g \in N$ then $(H_{(\alpha_1,\ldots,\alpha_k)})^g = H_{(\alpha_1^g,\alpha_2^g,\ldots,\alpha_k^g)}$, for all $\alpha_i \in \{1,\ldots,n\}$ and $1 \leq k \leq n$.
- Testing conjugacy of subgroups is hard in general, however there are various quick tests to show that two groups are NOT conjugate.

Normalisers of elementary abelian 2-groups

(From now on, joint work with Mun See Chang & Chris Jefferson).

Normalisers of elementary abelian 2-groups

(From now on, joint work with Mun See Chang & Chris Jefferson).

Fix $G = S_n$.

Normalisers of elementary abelian 2-groups

(From now on, joint work with Mun See Chang & Chris Jefferson).

Fix $G = S_n$. Consider subgroups
$H \leq E = \langle (1,2), (3,4), \ldots, \rangle$

Normalisers of elementary abelian 2-groups

(From now on, joint work with Mun See Chang & Chris Jefferson).

Fix $G = S_n$. Consider subgroups
$H \leq E = \langle (1,2), (3,4), \ldots, \rangle \cong C_2^{n/2} \leq S_n$.

Normalisers of elementary abelian 2-groups

(From now on, joint work with Mun See Chang & Chris Jefferson).

Fix $G = \mathsf{S}_n$. Consider subgroups
$H \leq E = \langle (1,2), (3,4), \ldots, \rangle \cong C_2^{n/2} \leq \mathsf{S}_n$.
Want to find $N = N_{\mathsf{S}_n}(H) \leq C_2 \wr \mathsf{S}_{n/2}$. (Assume $\mathrm{fix}(H) = \emptyset$).

Normalisers of elementary abelian 2-groups

(From now on, joint work with Mun See Chang & Chris Jefferson).

Fix $G = \mathsf{S}_n$. Consider subgroups
$H \leq E = \langle (1,2), (3,4), \ldots, \rangle \cong C_2^{n/2} \leq \mathsf{S}_n$.
Want to find $N = N_{\mathsf{S}_n}(H) \leq C_2 \wr \mathsf{S}_{n/2}$. (Assume $\mathrm{fix}(H) = \emptyset$).

- These are the groups which yield the lower bound in Pyber's count of subgroups of S_n.

Normalisers of elementary abelian 2-groups

(From now on, joint work with Mun See Chang & Chris Jefferson).

Fix $G = \mathsf{S}_n$. Consider subgroups
$H \leq E = \langle (1,2), (3,4), \ldots, \rangle \cong C_2^{n/2} \leq \mathsf{S}_n$.
Want to find $N = N_{\mathsf{S}_n}(H) \leq C_2 \wr \mathsf{S}_{n/2}$. (Assume $\mathrm{fix}(H) = \emptyset$).

- These are the groups which yield the lower bound in Pyber's count of subgroups of S_n.
- Similar methods work for subgroups of $C_p^{n/p}$ for all primes p.

Normalisers of elementary abelian 2-groups

(From now on, joint work with Mun See Chang & Chris Jefferson).

Fix $G = \mathsf{S}_n$. Consider subgroups
$H \leq E = \langle (1,2), (3,4), \ldots, \rangle \cong C_2^{n/2} \leq \mathsf{S}_n$.
Want to find $N = N_{\mathsf{S}_n}(H) \leq C_2 \wr \mathsf{S}_{n/2}$. (Assume $\mathrm{fix}(H) = \emptyset$).

- These are the groups which yield the lower bound in Pyber's count of subgroups of S_n.
- Similar methods work for subgroups of $C_p^{n/p}$ for all primes p.
- Methods will also be useful for groups H that have several orbits on which they act as C_p.

Normalisers of elementary abelian 2-groups

(From now on, joint work with Mun See Chang & Chris Jefferson).

Fix $G = \mathsf{S}_n$. Consider subgroups
$H \leq E = \langle (1,2), (3,4), \ldots, \rangle \cong C_2^{n/2} \leq \mathsf{S}_n$.
Want to find $N = N_{\mathsf{S}_n}(H) \leq C_2 \wr \mathsf{S}_{n/2}$. (Assume $\mathrm{fix}(H) = \emptyset$).

- These are the groups which yield the lower bound in Pyber's count of subgroups of S_n.
- Similar methods work for subgroups of $C_p^{n/p}$ for all primes p.
- Methods will also be useful for groups H that have several orbits on which they act as C_p.

$E \leq C_{\mathsf{S}_n}(H)$

Normalisers of elementary abelian 2-groups

(From now on, joint work with Mun See Chang & Chris Jefferson).

Fix $G = \mathsf{S}_n$. Consider subgroups
$H \leq E = \langle (1,2), (3,4), \ldots, \rangle \cong C_2^{n/2} \leq \mathsf{S}_n$.
Want to find $N = N_{\mathsf{S}_n}(H) \leq C_2 \wr \mathsf{S}_{n/2}$. (Assume $\mathrm{fix}(H) = \emptyset$).

- These are the groups which yield the lower bound in Pyber's count of subgroups of S_n.
- Similar methods work for subgroups of $C_p^{n/p}$ for all primes p.
- Methods will also be useful for groups H that have several orbits on which they act as C_p.

$E \leq C_{\mathsf{S}_n}(H)$, so need to find generators for $N/E \leq \mathsf{S}_{n/2}$.

Normalisers of elementary abelian 2-groups

(From now on, joint work with Mun See Chang & Chris Jefferson).

Fix $G = \mathsf{S}_n$. Consider subgroups
$H \leq E = \langle (1,2), (3,4), \ldots, \rangle \cong C_2^{n/2} \leq \mathsf{S}_n$.
Want to find $N = N_{\mathsf{S}_n}(H) \leq C_2 \wr \mathsf{S}_{n/2}$. (Assume $\mathrm{fix}(H) = \emptyset$).

- These are the groups which yield the lower bound in Pyber's count of subgroups of S_n.
- Similar methods work for subgroups of $C_p^{n/p}$ for all primes p.
- Methods will also be useful for groups H that have several orbits on which they act as C_p.

$E \leq C_{\mathsf{S}_n}(H)$, so need to find generators for $N/E \leq \mathsf{S}_{n/2}$.
Identify E with $\mathbb{F}_2^{n/2}$

Normalisers of elementary abelian 2-groups

(From now on, joint work with Mun See Chang & Chris Jefferson).

Fix $G = \mathsf{S}_n$. Consider subgroups
$H \leq E = \langle (1,2), (3,4), \ldots, \rangle \cong C_2^{n/2} \leq \mathsf{S}_n$.
Want to find $N = N_{\mathsf{S}_n}(H) \leq C_2 \wr \mathsf{S}_{n/2}$. (Assume $\mathrm{fix}(H) = \emptyset$).

- These are the groups which yield the lower bound in Pyber's count of subgroups of S_n.
- Similar methods work for subgroups of $C_p^{n/p}$ for all primes p.
- Methods will also be useful for groups H that have several orbits on which they act as C_p.

$E \leq C_{\mathsf{S}_n}(H)$, so need to find generators for $N/E \leq \mathsf{S}_{n/2}$.
Identify E with $\mathbb{F}_2^{n/2}$, and H with a subspace of E.

Normalisers of elementary abelian 2-groups

(From now on, joint work with Mun See Chang & Chris Jefferson).

Fix $G = \mathsf{S}_n$. Consider subgroups
$H \leq E = \langle (1,2), (3,4), \ldots, \rangle \cong C_2^{n/2} \leq \mathsf{S}_n$.
Want to find $N = N_{\mathsf{S}_n}(H) \leq C_2 \wr \mathsf{S}_{n/2}$. (Assume $\mathrm{fix}(H) = \emptyset$).

- These are the groups which yield the lower bound in Pyber's count of subgroups of S_n.
- Similar methods work for subgroups of $C_p^{n/p}$ for all primes p.
- Methods will also be useful for groups H that have several orbits on which they act as C_p.

$E \leq C_{\mathsf{S}_n}(H)$, so need to find generators for $N/E \leq \mathsf{S}_{n/2}$.
Identify E with $\mathbb{F}_2^{n/2}$, and H with a subspace of E.
Describe H by a $k \times n/2$ matrix M_H, where $|H| = 2^k$.

Normalisers of elementary abelian 2-groups

(From now on, joint work with Mun See Chang & Chris Jefferson).

Fix $G = \mathsf{S}_n$. Consider subgroups
$H \leq E = \langle (1,2), (3,4), \ldots, \rangle \cong C_2^{n/2} \leq \mathsf{S}_n$.
Want to find $N = N_{\mathsf{S}_n}(H) \leq C_2 \wr \mathsf{S}_{n/2}$. (Assume $\mathrm{fix}(H) = \emptyset$).

- These are the groups which yield the lower bound in Pyber's count of subgroups of S_n.
- Similar methods work for subgroups of $C_p^{n/p}$ for all primes p.
- Methods will also be useful for groups H that have several orbits on which they act as C_p.

$E \leq C_{\mathsf{S}_n}(H)$, so need to find generators for $N/E \leq \mathsf{S}_{n/2}$.
Identify E with $\mathbb{F}_2^{n/2}$, and H with a subspace of E.
Describe H by a $k \times n/2$ matrix M_H, where $|H| = 2^k$.

An example

$$H = \langle (1,7)(2,8)(5,11), (1,7)(2,8)(3,9)(6,12), (2,8)(3,9)(4,10) \rangle$$

An example

$$H = \langle (1,7)(2,8)(5,11), (1,7)(2,8)(3,9)(6,12), (2,8)(3,9)(4,10) \rangle$$

Then $M_H = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}$

An example

$$H = \langle (1,7)(2,8)(5,11), (1,7)(2,8)(3,9)(6,12), (2,8)(3,9)(4,10) \rangle$$

Then $M_H = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}.$

An example

$$H = \langle (1,7)(2,8)(5,11), (1,7)(2,8)(3,9)(6,12), (2,8)(3,9)(4,10) \rangle$$

Then $M_H = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$.

Reduce search space by:

An example

$$H = \langle (1,7)(2,8)(5,11), (1,7)(2,8)(3,9)(6,12), (2,8)(3,9)(4,10) \rangle$$

Then $M_H = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}.$

Reduce search space by:

- Once the image of $k = 3$ points in $\{1, \ldots, n/2\}$ are specified under $g \in S_n$

An example

$$H = \langle (1,7)(2,8)(5,11), (1,7)(2,8)(3,9)(6,12), (2,8)(3,9)(4,10) \rangle$$

Then $M_H = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$.

Reduce search space by:

- Once the image of $k = 3$ points in $\{1, \ldots, n/2\}$ are specified under $g \in S_n$, we know x^g for all $x \in H$ (up to E).

An example

$$H = \langle (1,7)(2,8)(5,11), (1,7)(2,8)(3,9)(6,12), (2,8)(3,9)(4,10) \rangle$$

Then $M_H = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$.

Reduce search space by:

- Once the image of $k = 3$ points in $\{1, \ldots, n/2\}$ are specified under $g \in S_n$, we know x^g for all $x \in H$ (up to E). So we can decide whether $g \in N$.

An example

$$H = \langle (1,7)(2,8)(5,11), (1,7)(2,8)(3,9)(6,12), (2,8)(3,9)(4,10) \rangle$$

Then $M_H = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$.

Reduce search space by:

- Once the image of $k = 3$ points in $\{1, \ldots, n/2\}$ are specified under $g \in S_n$, we know x^g for all $x \in H$ (up to E). So we can decide whether $g \in N$.
- N permutes sets of linearly dependent columns of M_H.

An example

$$H = \langle (1,7)(2,8)(5,11), (1,7)(2,8)(3,9)(6,12), (2,8)(3,9)(4,10) \rangle$$

Then $M_H = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$.

Reduce search space by:

- Once the image of $k = 3$ points in $\{1, \ldots, n/2\}$ are specified under $g \in S_n$, we know x^g for all $x \in H$ (up to E). So we can decide whether $g \in N$.
- N permutes sets of linearly dependent columns of M_H.
- For all $\alpha_i \in \{1, \ldots, n\}$, point stabilisers $H_{(\alpha_1, \alpha_2, \ldots, \alpha_i)}$

An example

$$H = \langle (1,7)(2,8)(5,11), (1,7)(2,8)(3,9)(6,12), (2,8)(3,9)(4,10) \rangle$$

Then $M_H = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$.

Reduce search space by:

- Once the image of $k = 3$ points in $\{1, \ldots, n/2\}$ are specified under $g \in S_n$, we know x^g for all $x \in H$ (up to E). So we can decide whether $g \in N$.
- N permutes sets of linearly dependent columns of M_H.
- For all $\alpha_i \in \{1, \ldots, n\}$, point stabilisers $H_{(\alpha_1, \alpha_2, \ldots, \alpha_i)}$ can be efficiently calculated from M_H.

An example

$$H = \langle (1,7)(2,8)(5,11), (1,7)(2,8)(3,9)(6,12), (2,8)(3,9)(4,10) \rangle$$

Then $M_H = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$.

Reduce search space by:

- Once the image of $k = 3$ points in $\{1, \ldots, n/2\}$ are specified under $g \in S_n$, we know x^g for all $x \in H$ (up to E). So we can decide whether $g \in N$.
- N permutes sets of linearly dependent columns of M_H.
- For all $\alpha_i \in \{1, \ldots, n\}$, point stabilisers $H_{(\alpha_1, \alpha_2, \ldots, \alpha_i)}$ can be efficiently calculated from M_H.
- Let K_H be a row rank $(n/2 - k)$ matrix whose nullspace is H.

An example

$$H = \langle (1,7)(2,8)(5,11), (1,7)(2,8)(3,9)(6,12), (2,8)(3,9)(4,10) \rangle$$

Then $M_H = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$.

Reduce search space by:

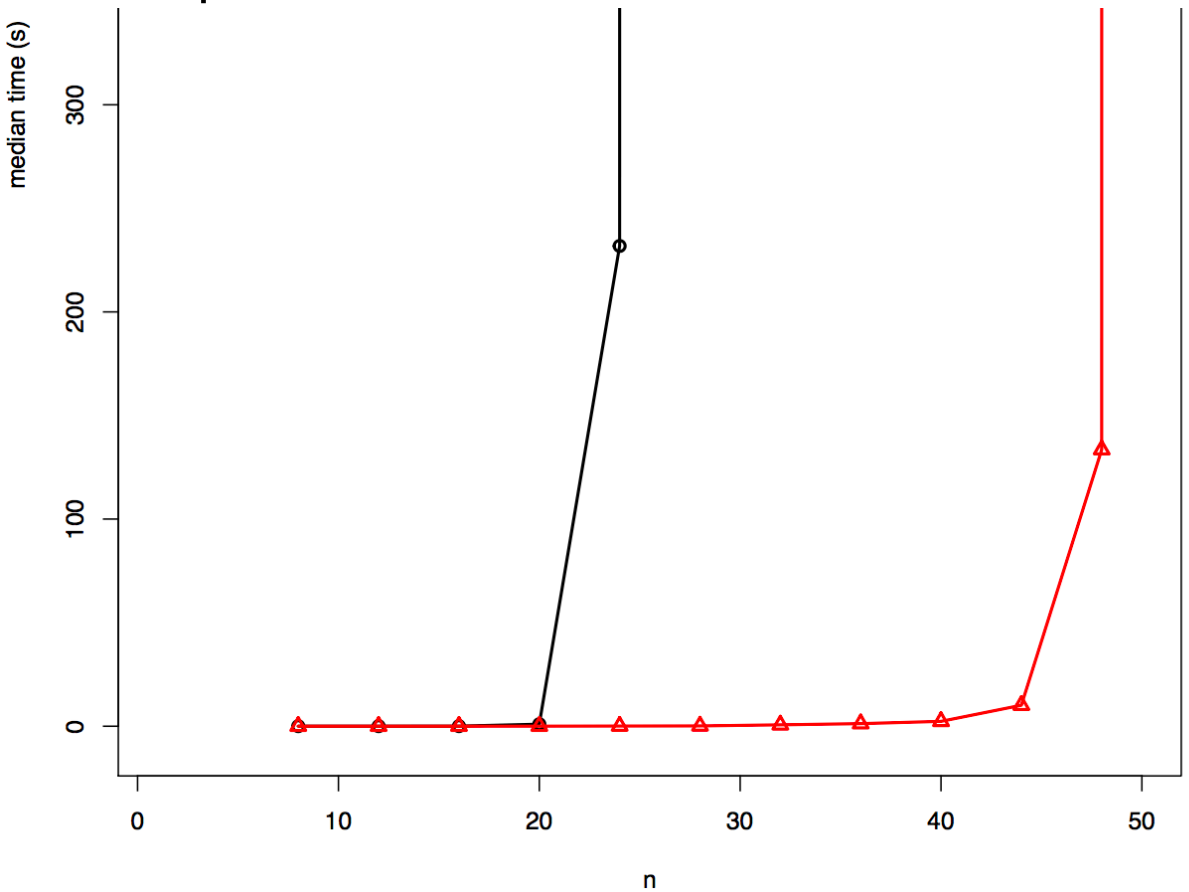
- Once the image of $k = 3$ points in $\{1, \ldots, n/2\}$ are specified under $g \in S_n$, we know x^g for all $x \in H$ (up to E). So we can decide whether $g \in N$.
- N permutes sets of linearly dependent columns of M_H.
- For all $\alpha_i \in \{1, \ldots, n\}$, point stabilisers $H_{(\alpha_1, \alpha_2, \ldots, \alpha_i)}$ can be efficiently calculated from M_H.
- Let K_H be a row rank $(n/2 - k)$ matrix whose nullspace is H. Then N also acts naturally on columns of K_H, and similar observations apply.

Some timings

Our methods for the normaliser of subgroups of $C_p^{n/p}$ are still worst case exponential.

Some timings

Our methods for the normaliser of subgroups of $C_p^{n/p}$ are still worst case exponential.



Red = our algorithm

Black = standard algorithm.

Time recorded is median over 20 random instances, with a 600 second timeout.