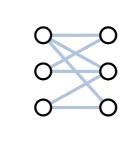
Exponential Time Algorithms

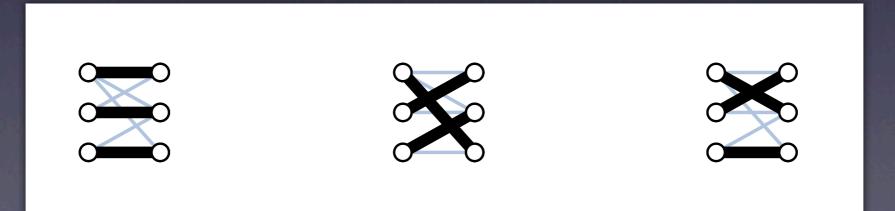
Thore Husfeldt

IT University of Copenhagen Lund University

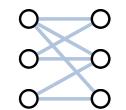
Perfect Matchings

Perfect Matchings in Bipartite Graphs





onsdag 12 oktober 11



1 1 1

1 1 0

0 1 1

1 1 1

1 1 0

1

0

1



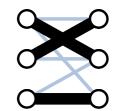
1 1 1

0

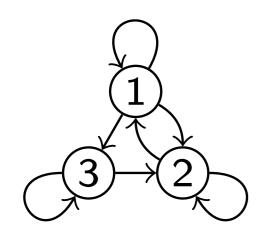
1

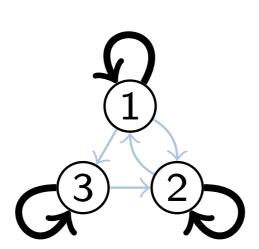
1 1

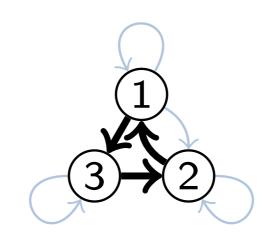
0 1

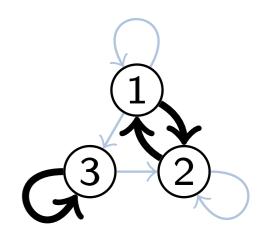


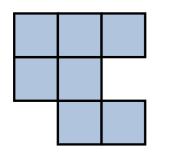
 $\begin{array}{cccc} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{array}$

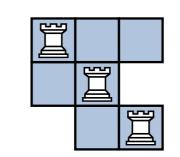


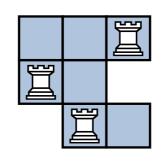


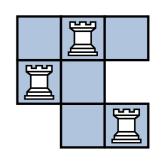






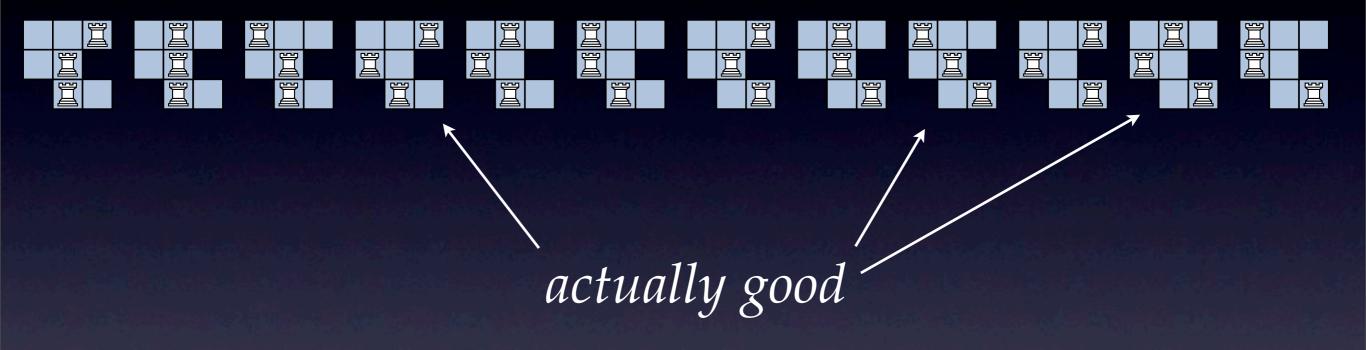




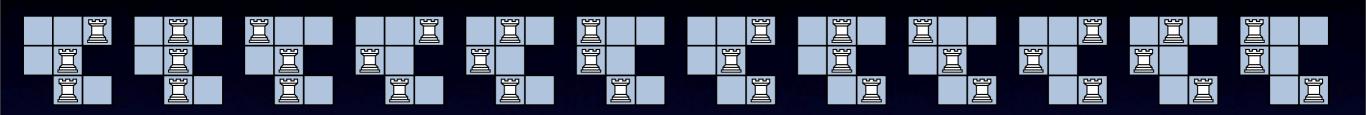


onsdag 12 oktober 11

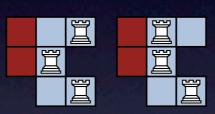






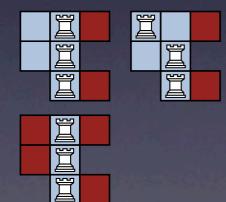


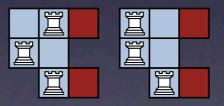


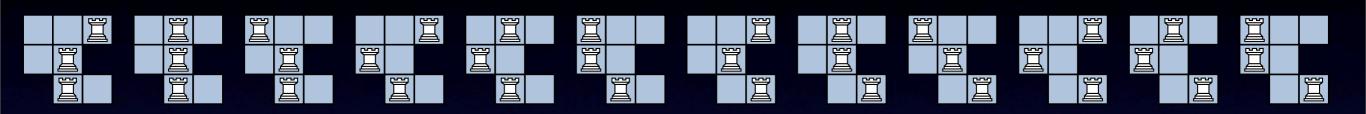


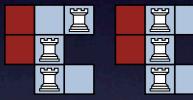


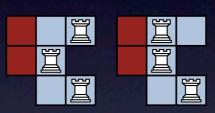






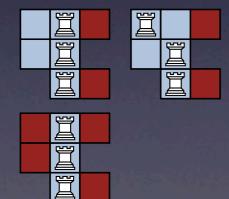










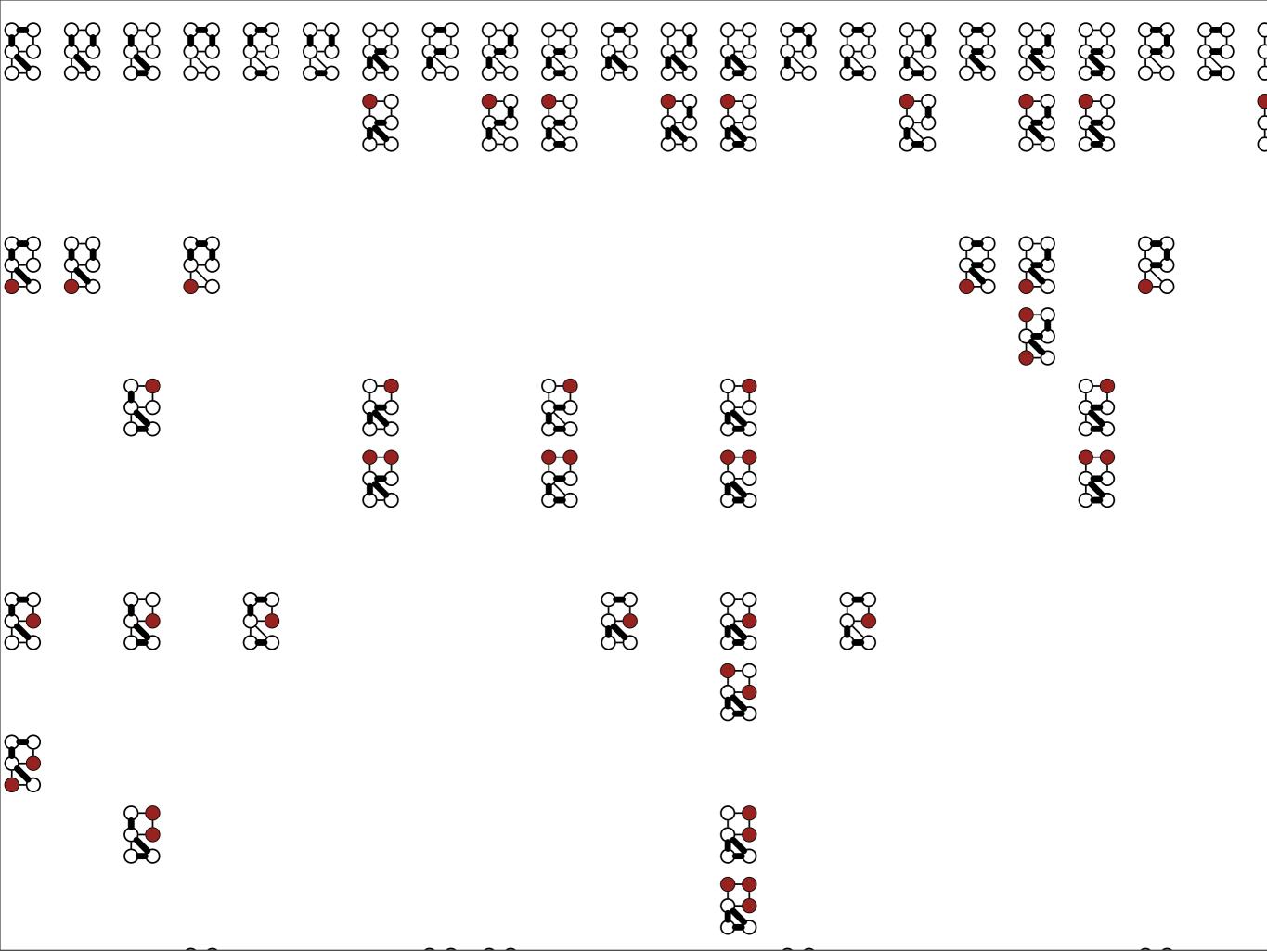


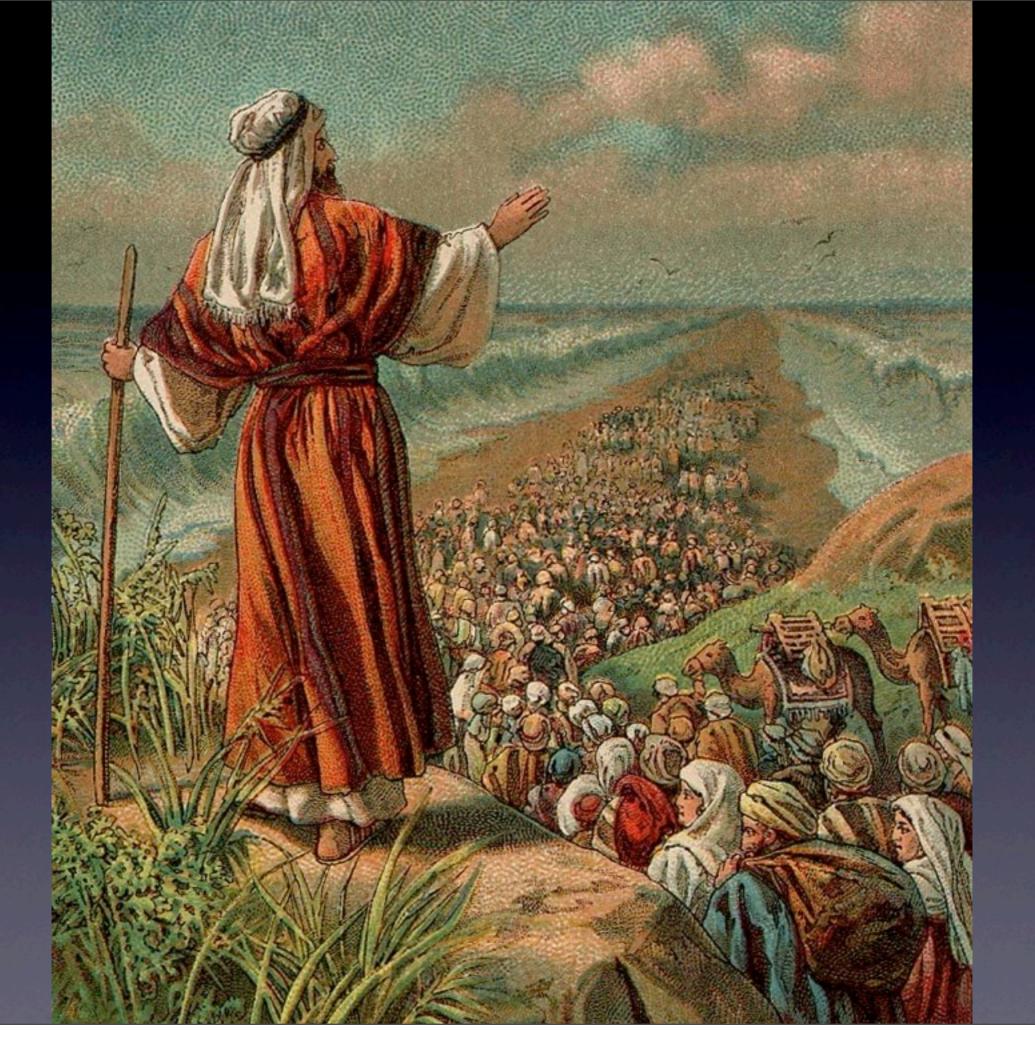




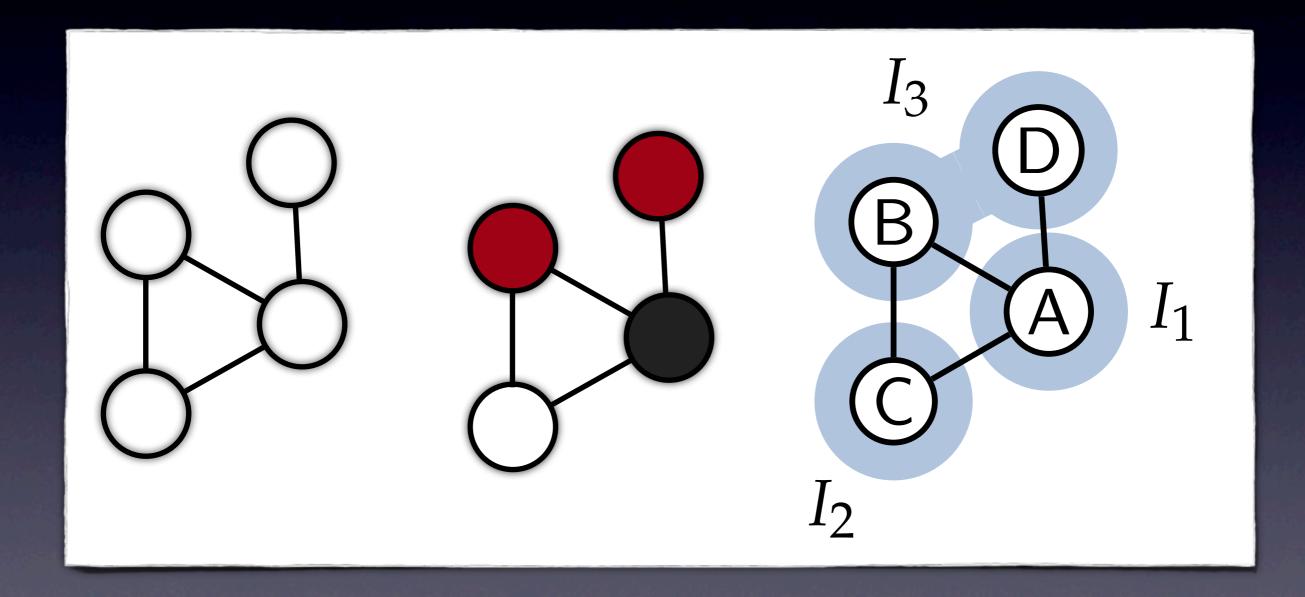
Ryser

													•••				
								5									

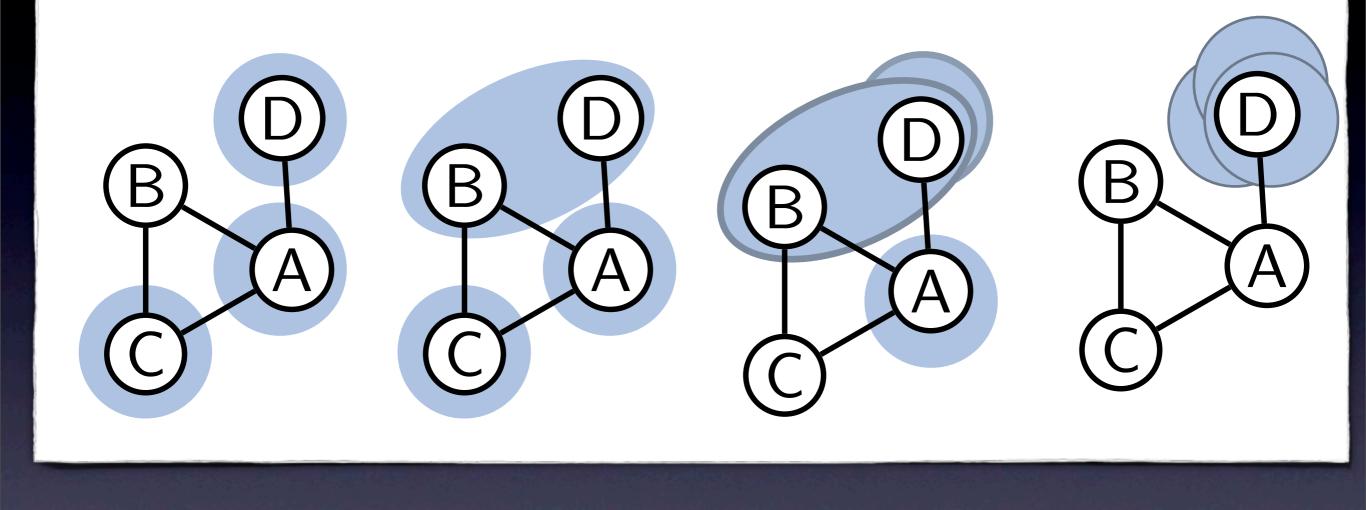




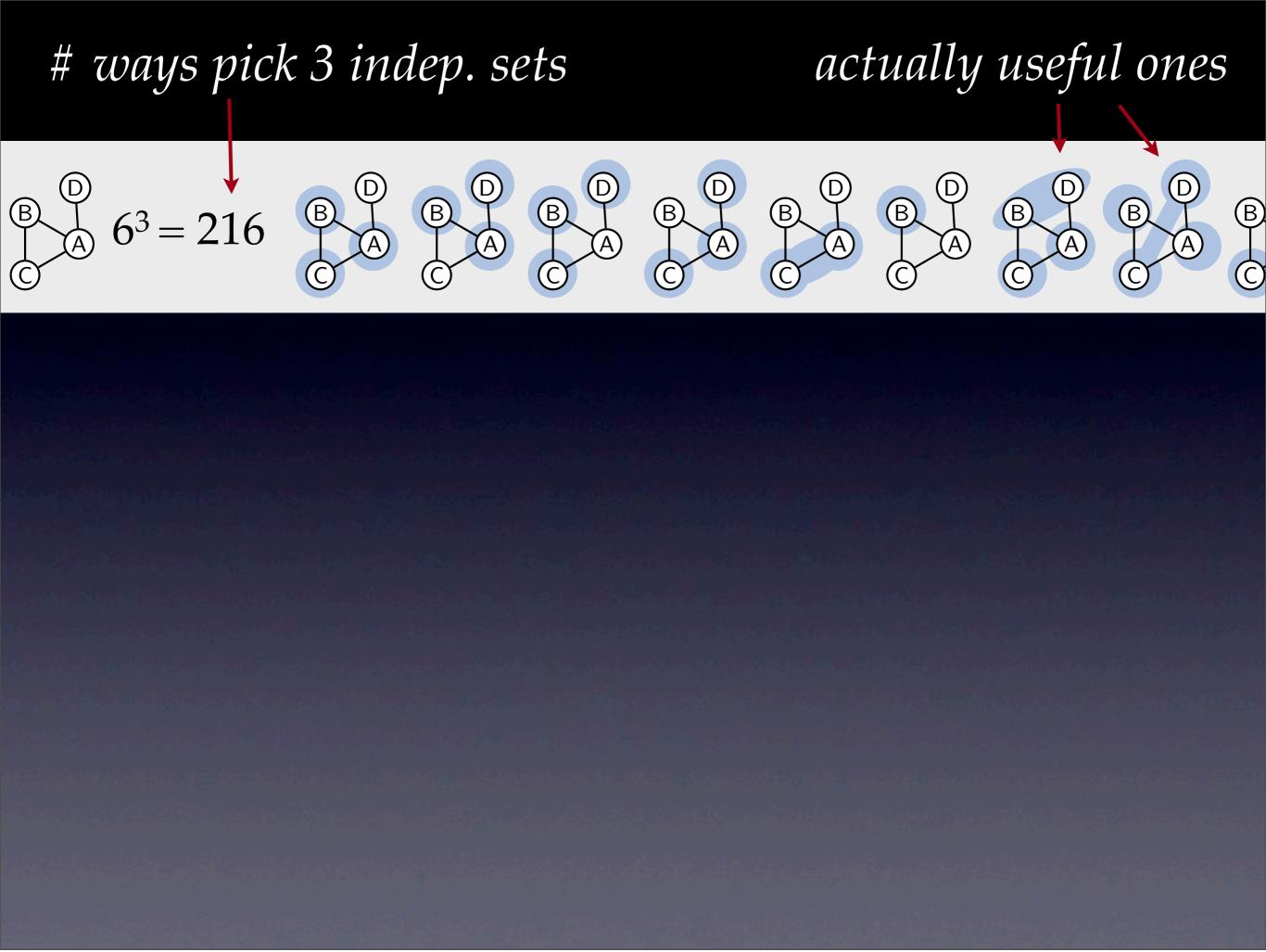
Vertex colouring

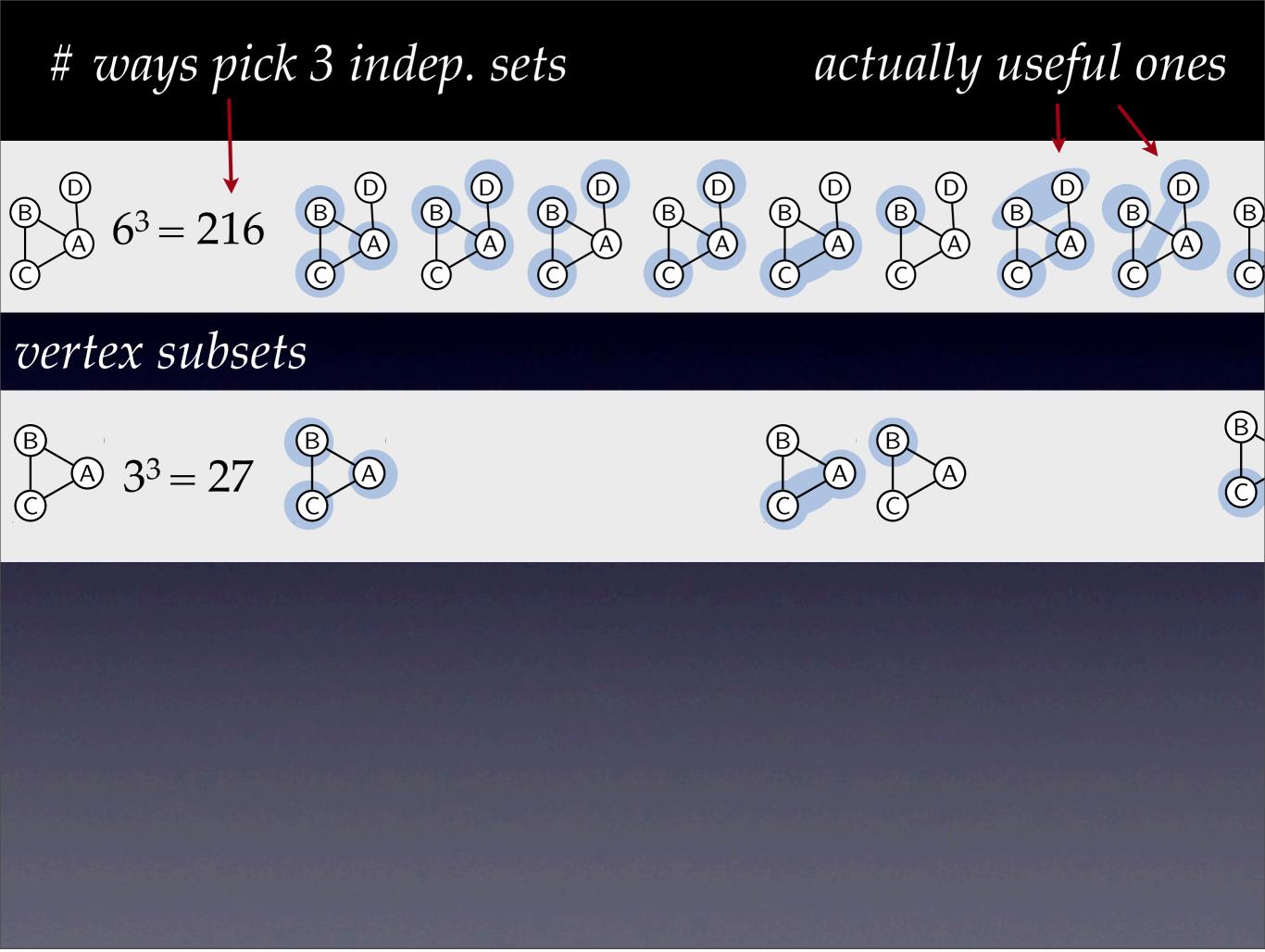


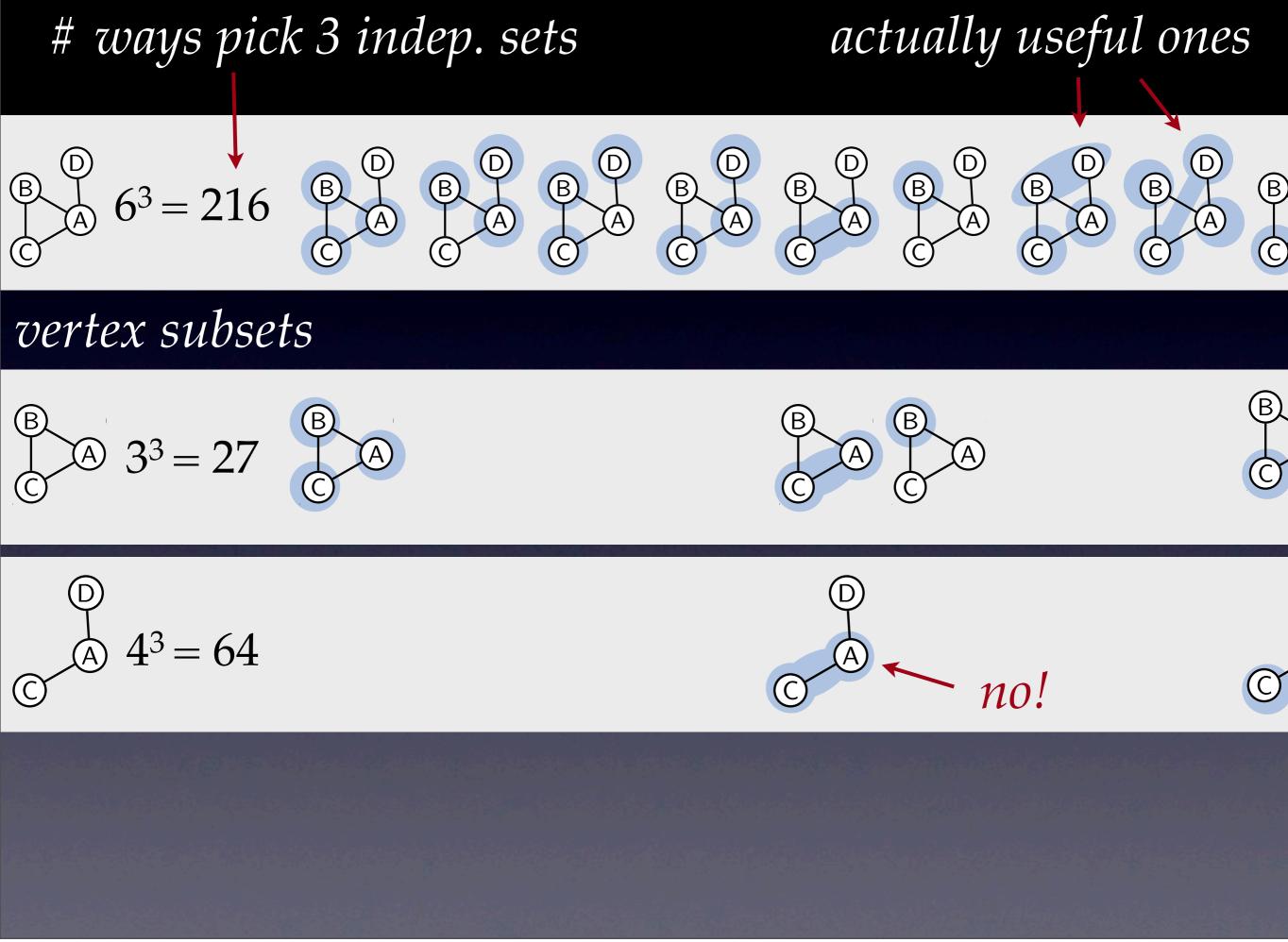
Picking 3 independent sets

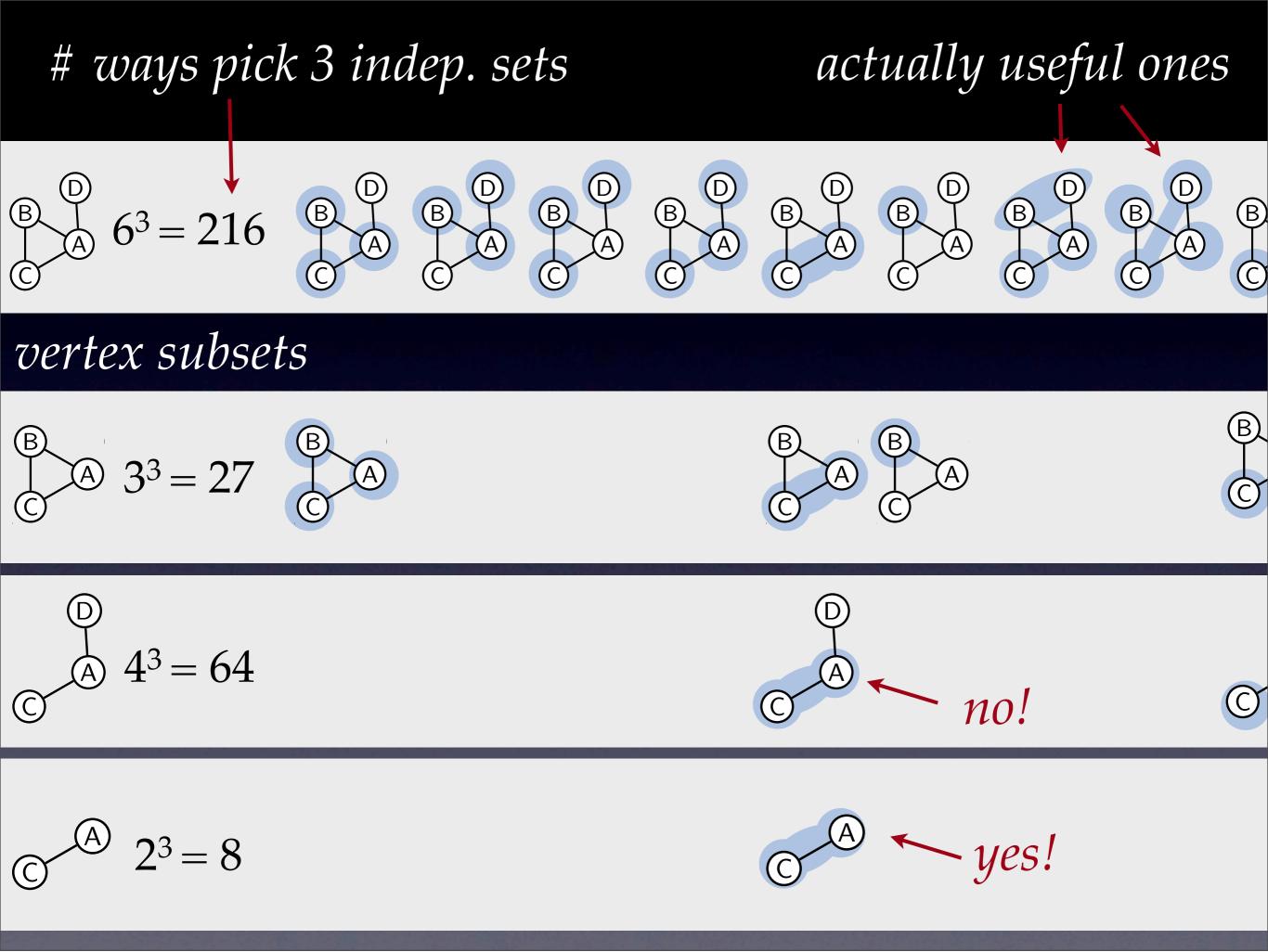


A	В	C	D	AB	AC	AD	BC	BD	CD	ABC	ABD	ACD	BCD	ABCD
Ι	I	Ι	I	0	0	0	0	I	I	0	0	0	0	0

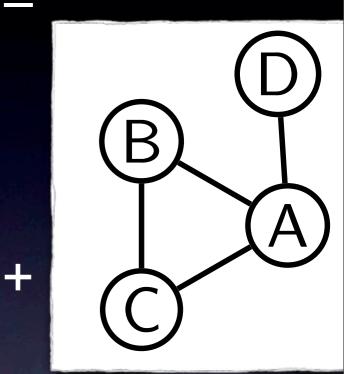


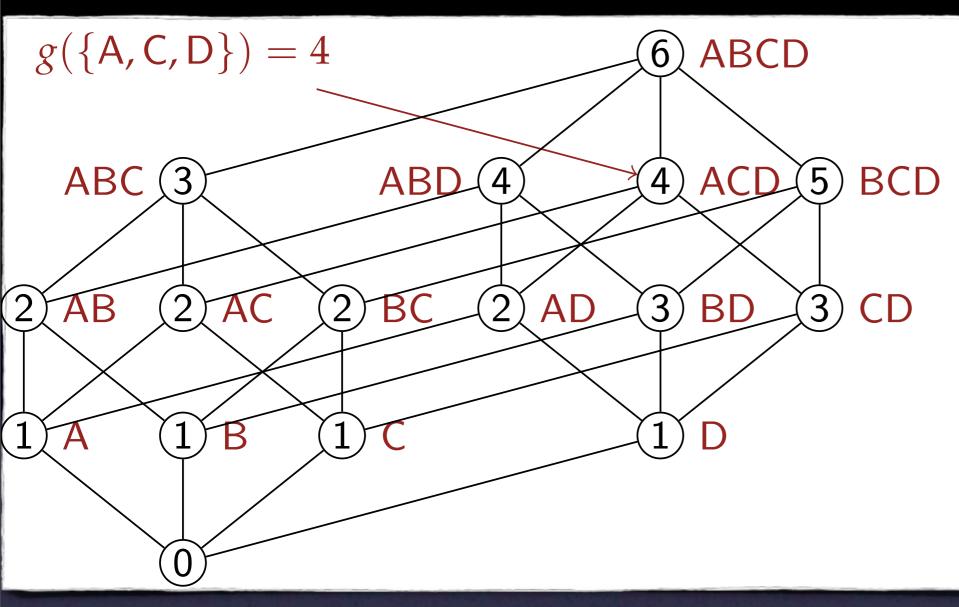


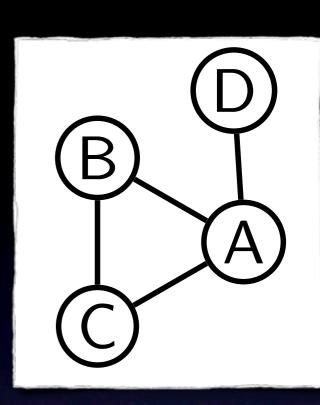




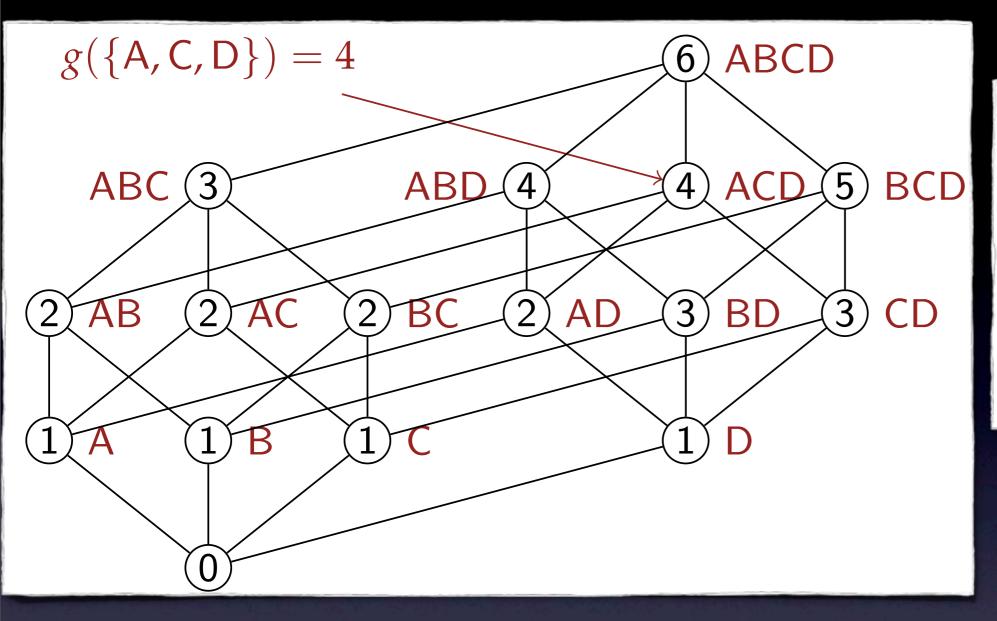
Vertex subset <i>S</i>	# indep. subsets, $g(S)$	$(g(S))^{3}$	
A	1	1	
В	1	1	
С	1	1	
		1	
AB	2	8	
AC	2	8	
AD	2	8	
BC	2	8	
BD	3	27	
CD	3	27	
ABC	3	27	
ABD	4	64	
ACD	4	64	
BCD	5	125	
ABCD	6	216	

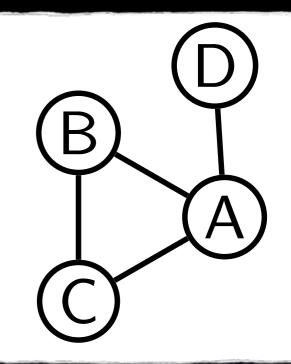


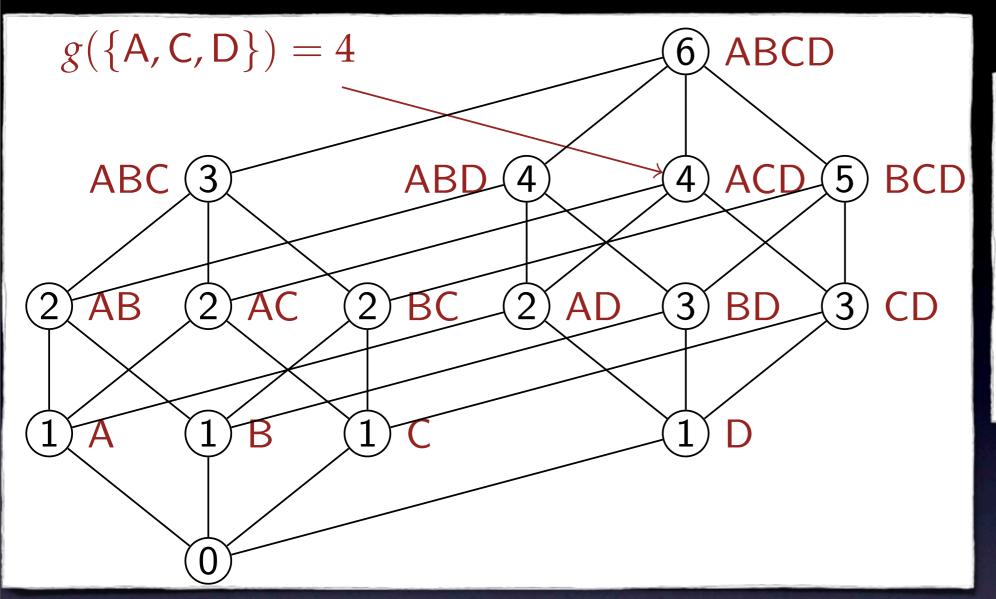




Vertex subset S	# indep.subsets, g(S)	(g(S)) ³
A	1	1
В	1	1
C	1 1 1	1
D	1	1
AB	2	8
AC	2	8
AD	2	8
BC	2	8
BD	3	27
CD	3	27
ABC	3	27
ABD	4	64
ACD	4	64
BCD	5	125
ABCD	6	216

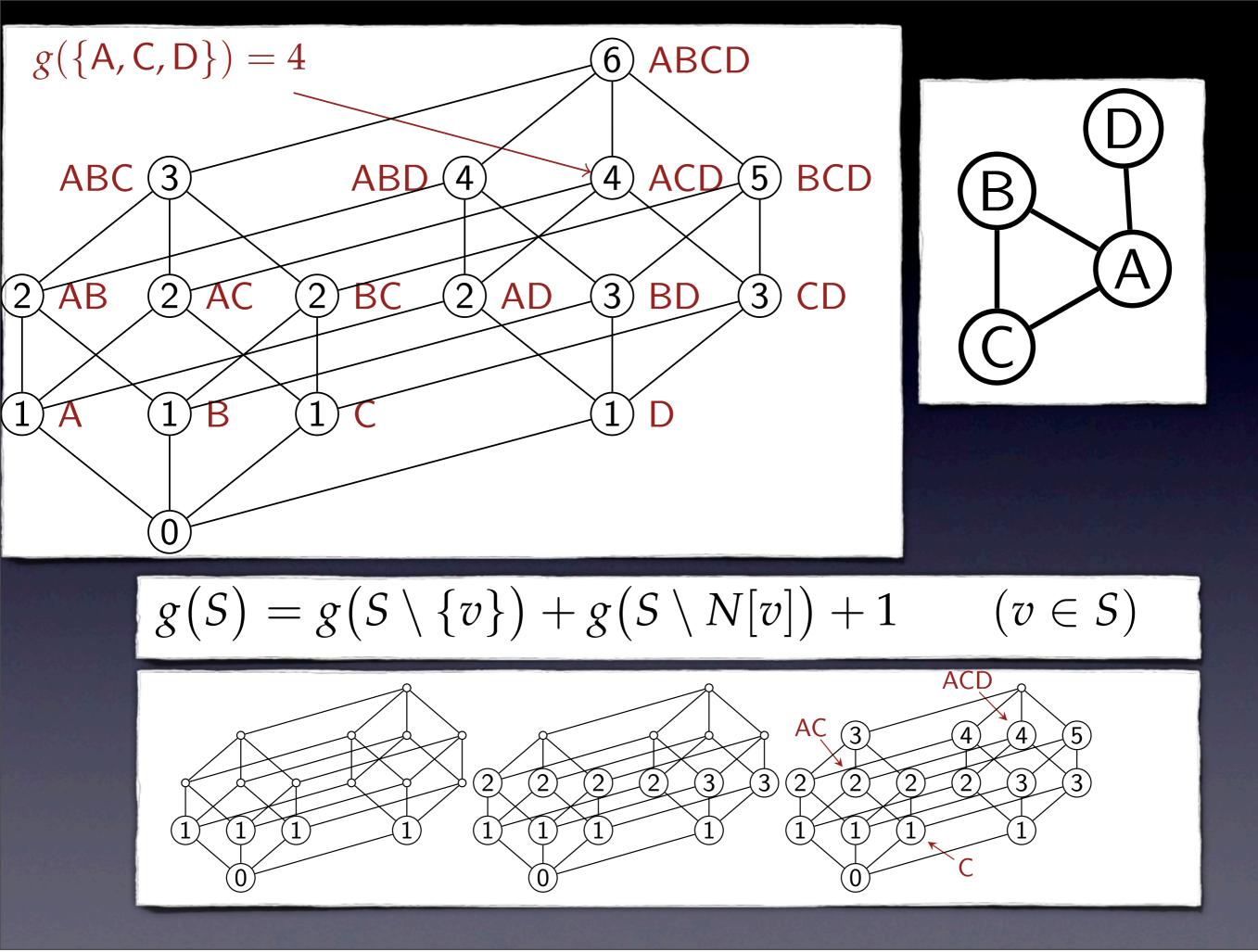






$$\sum_{S \subseteq N} 2^{|S|} = \sum_{i=1}^{n} \binom{n}{i} 2^{i} = 3^{n}$$

polynomial space



Graph colouring Compute $\sum_{S \subseteq N} (-1)^{n-|S|} (g(S))^k$

 $O^*(3^n)$ time polynomial space

$O^*(2^n)$ time $O^*(2^n)$ space

Koivisto

Björklund

Exponential Time Hypothesis

Exponential Time Hypothesis

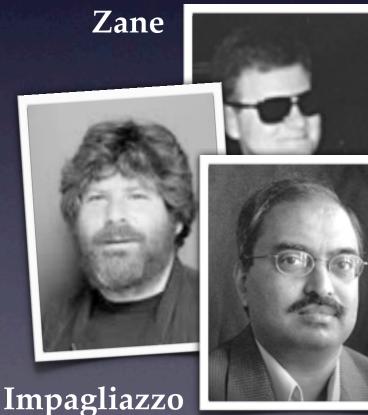
Hertli



Exponential Time Hypothesis

Hertli



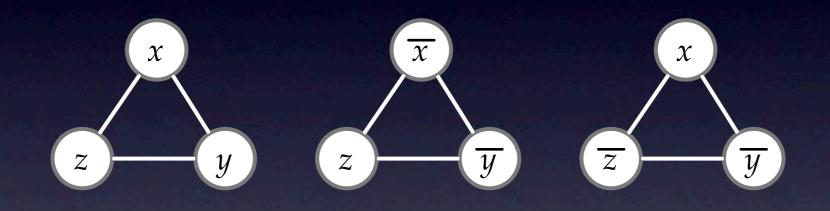


Paturi

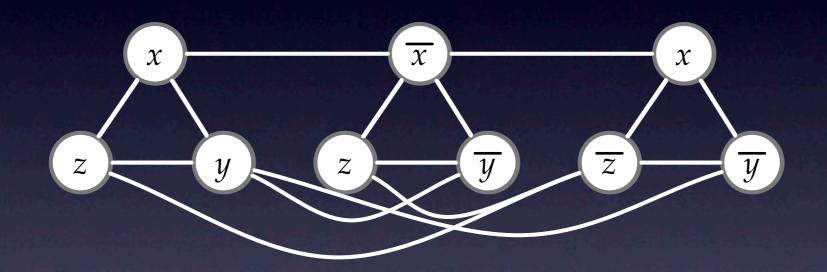
Can't do 3-Sat in *time* exp(*o*(*n*))

onsdag 12 oktober 11

$(x \lor y \lor z) \land (\overline{x} \lor \overline{y} \lor z) \land (x \lor \overline{y} \lor \overline{z})$



$(x \lor y \lor z) \land (\overline{x} \lor \overline{y} \lor z) \land (x \lor \overline{y} \lor \overline{z})$



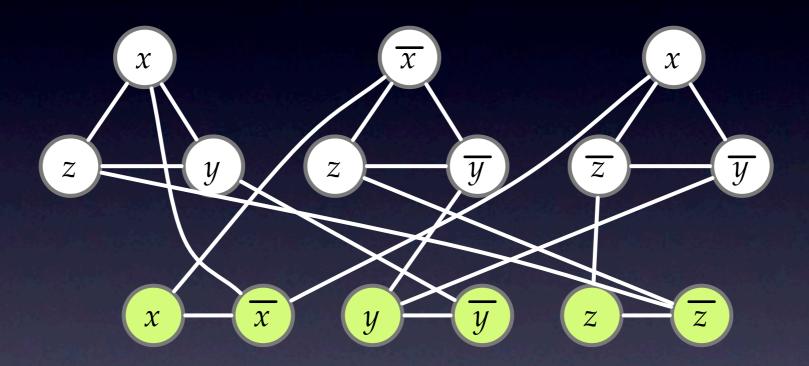
3m=O(n³) verts O(m²) edges

n vars m clauses

exp(o(n)) alg for 3-SAT

 $exp(o(n^{1/3}))$ alg for I.S. $exp(o(m^{1/2}))$ alg for I.S.

$(x \lor y \lor z) \land (\overline{x} \lor \overline{y} \lor z) \land (x \lor \overline{y} \lor \overline{z})$



3m=O(n³) verts O(m) edges

n vars m clauses

> $exp(o(n^{1/3}))$ alg for I.S. exp(o(m)) alg for I.S.

exp(o(n)) alg for 3-SAT

Independent set n vertices m edges

Clique n vertices m edges

 1.1888^{n}

 1.1888^{n}

 C^m

 $2^{\sqrt{m}\log n}$

Sparsification

Hitting Set

Hitting Set

Hitting Set

4 (5) 6 (7) 8

2

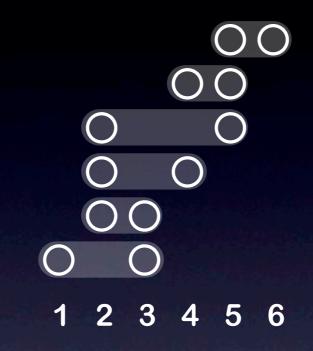
3

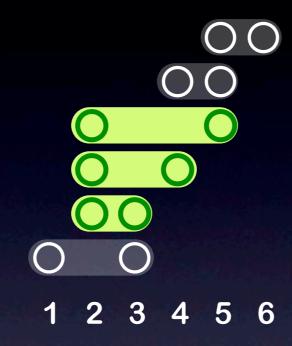
1

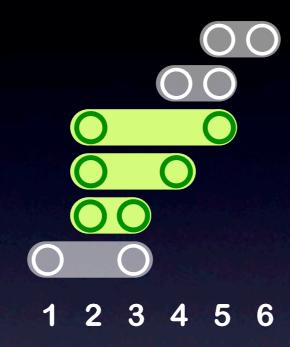
 $\sim n^2 sets$

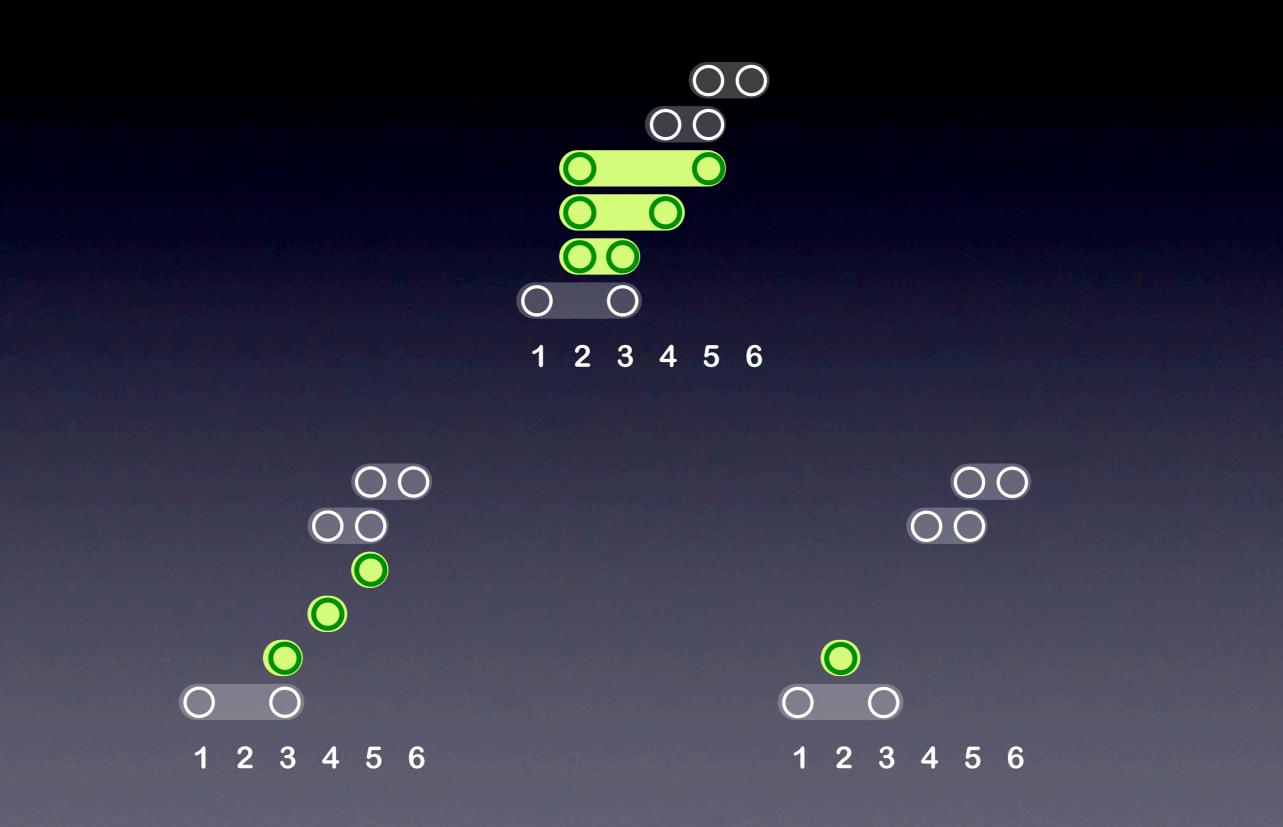
1 2 3 4 5 6 t

element of high "degree"



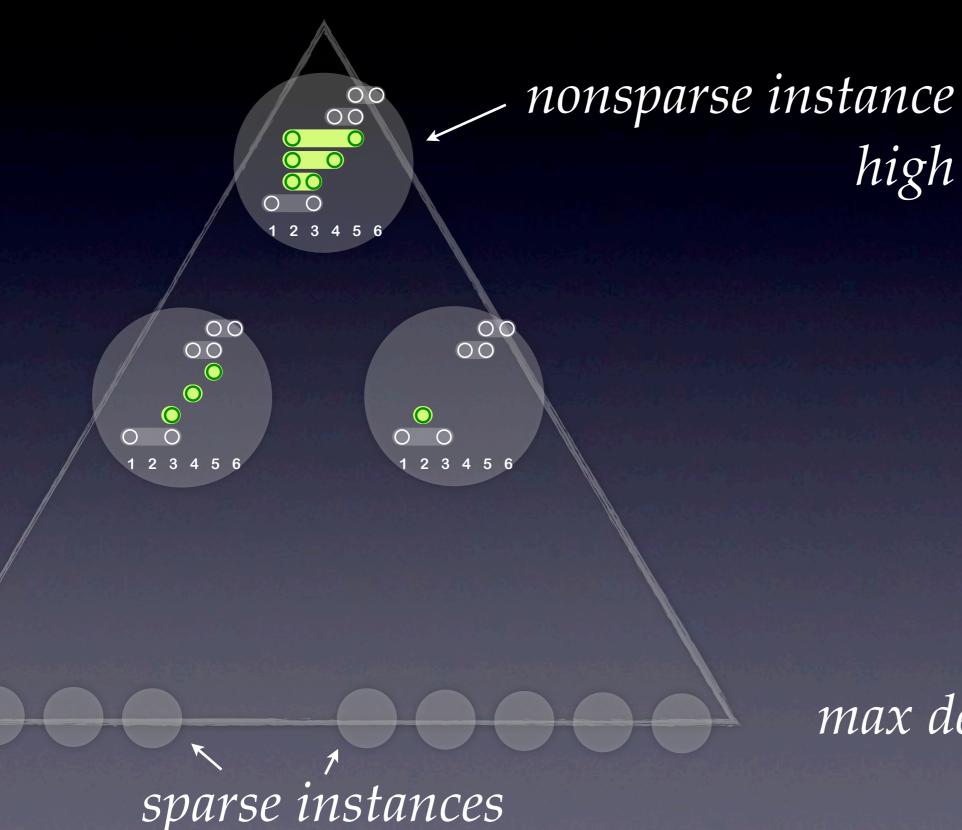






high degree

max degree $r \cdot n$





$exp(o(n)) \cdot 2^{n} = exp(n)$



$exp(o(n)) \cdot exp(H(1/r)n) = exp(o(n))$

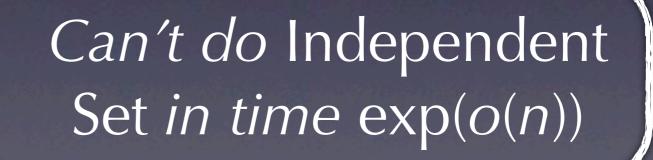
C(n,1) +...+ C(n,n/r) leaves

Exponential Time Hypothesis

Can't do 3-Sat in time exp(o(n))



Can't do 18-Sat *in time* exp(*o*(*m*))



Paturi

Impagliazzo

Why No Dependency on # Colours is Surprising

CSP(q,2)

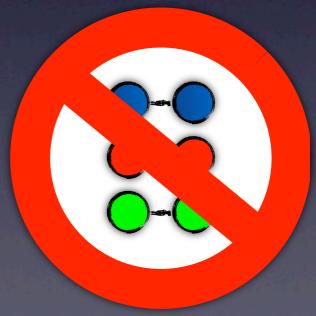
q states, pairwise constraints

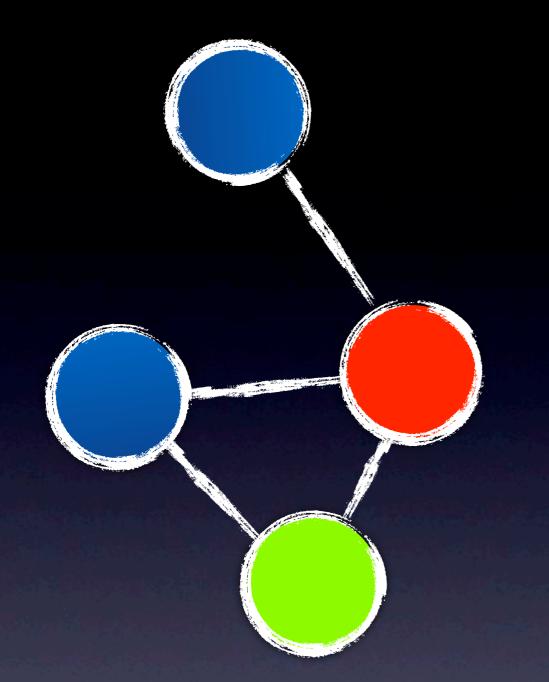


Traxler

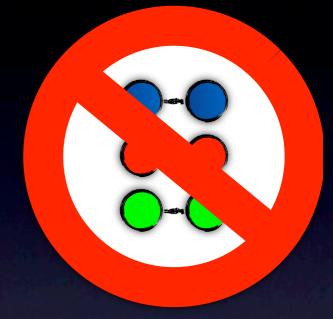


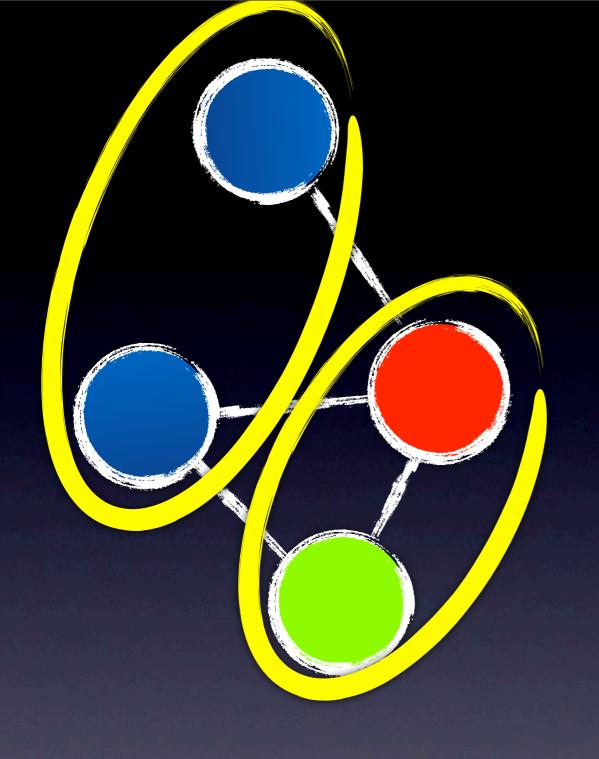
constraints



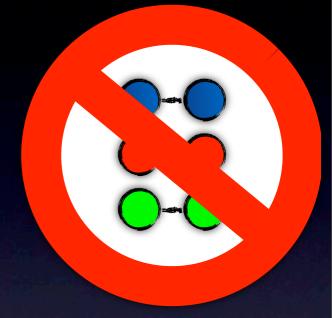


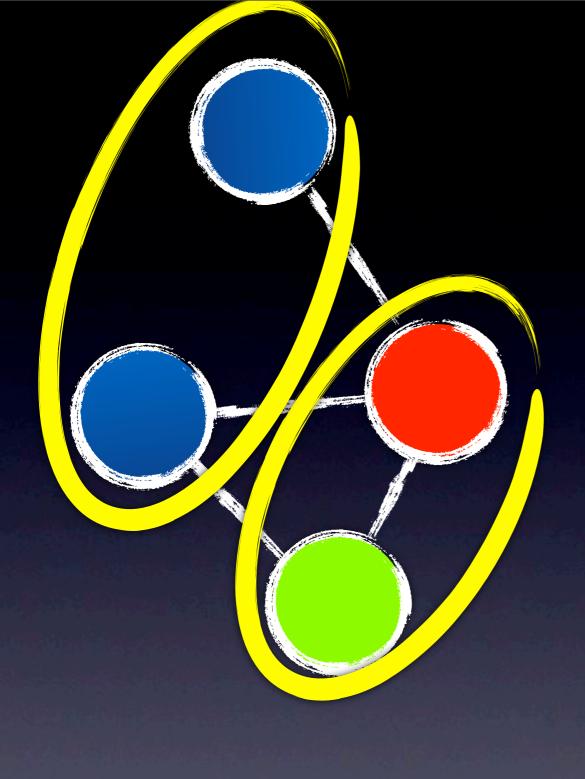
constraints

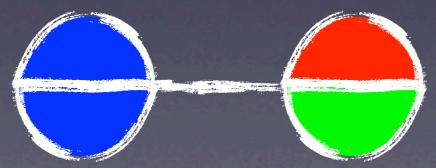




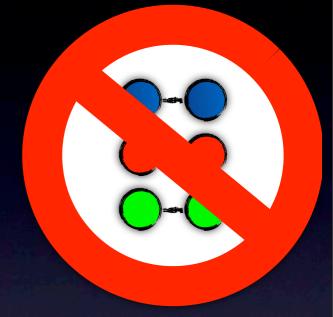
constraints

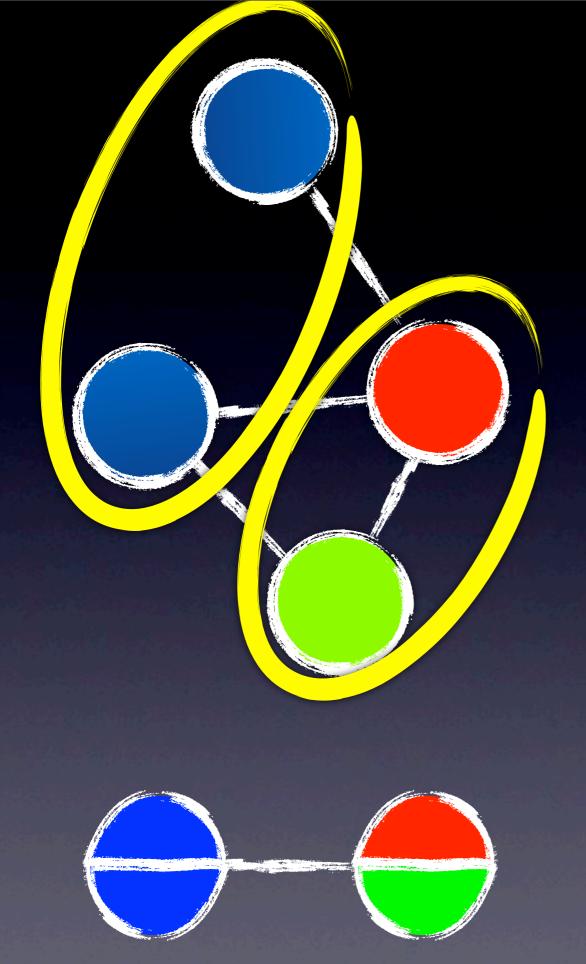






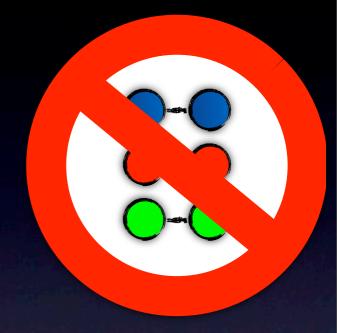
constraints

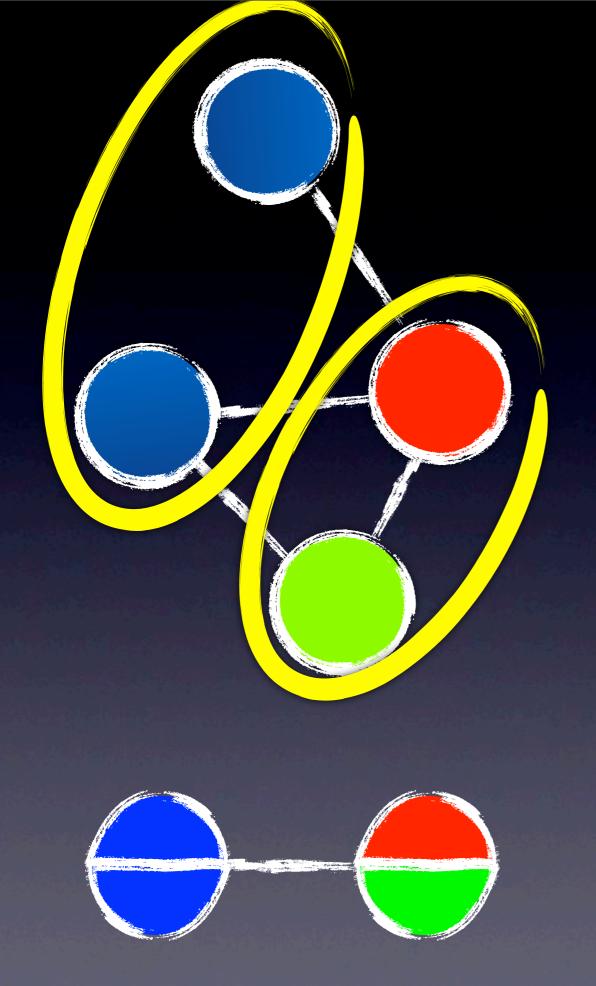




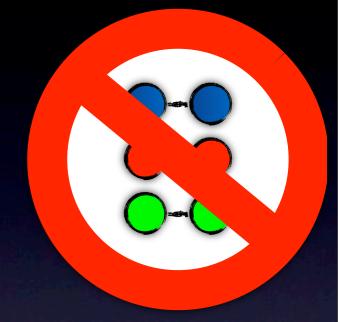


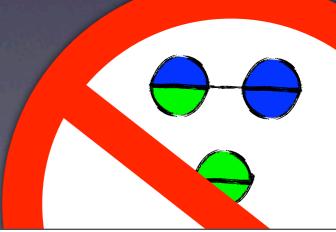
constraints





constraints

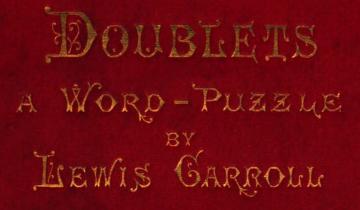




n verts d states n/2 verts d² states

Must have $d^n = (d^2)^{n/2}$

Path through specified vertices



DEAR VANITY,—Just a year ago last Christmas, two young ladies —smarting under that sorest scourge of feminine humanity, the having "nothing to do"—besought me to send them "some riddles." But riddles I had none at hand, and therefore set myself in large some

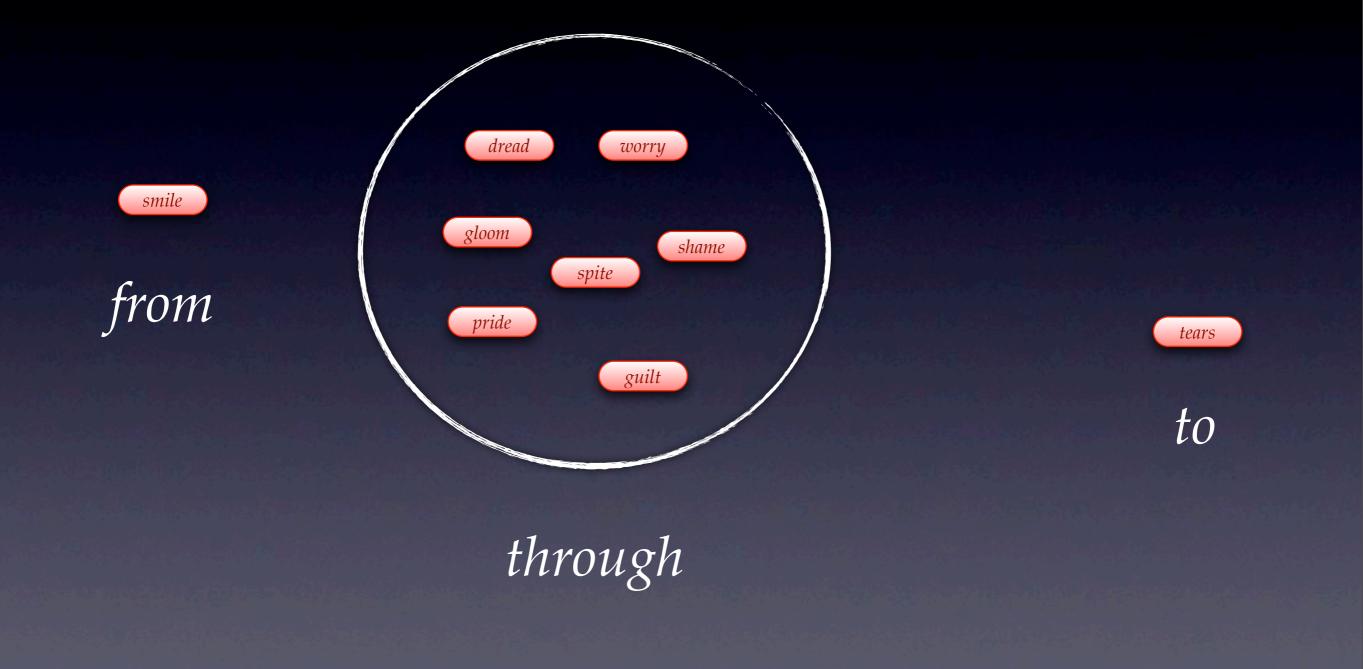


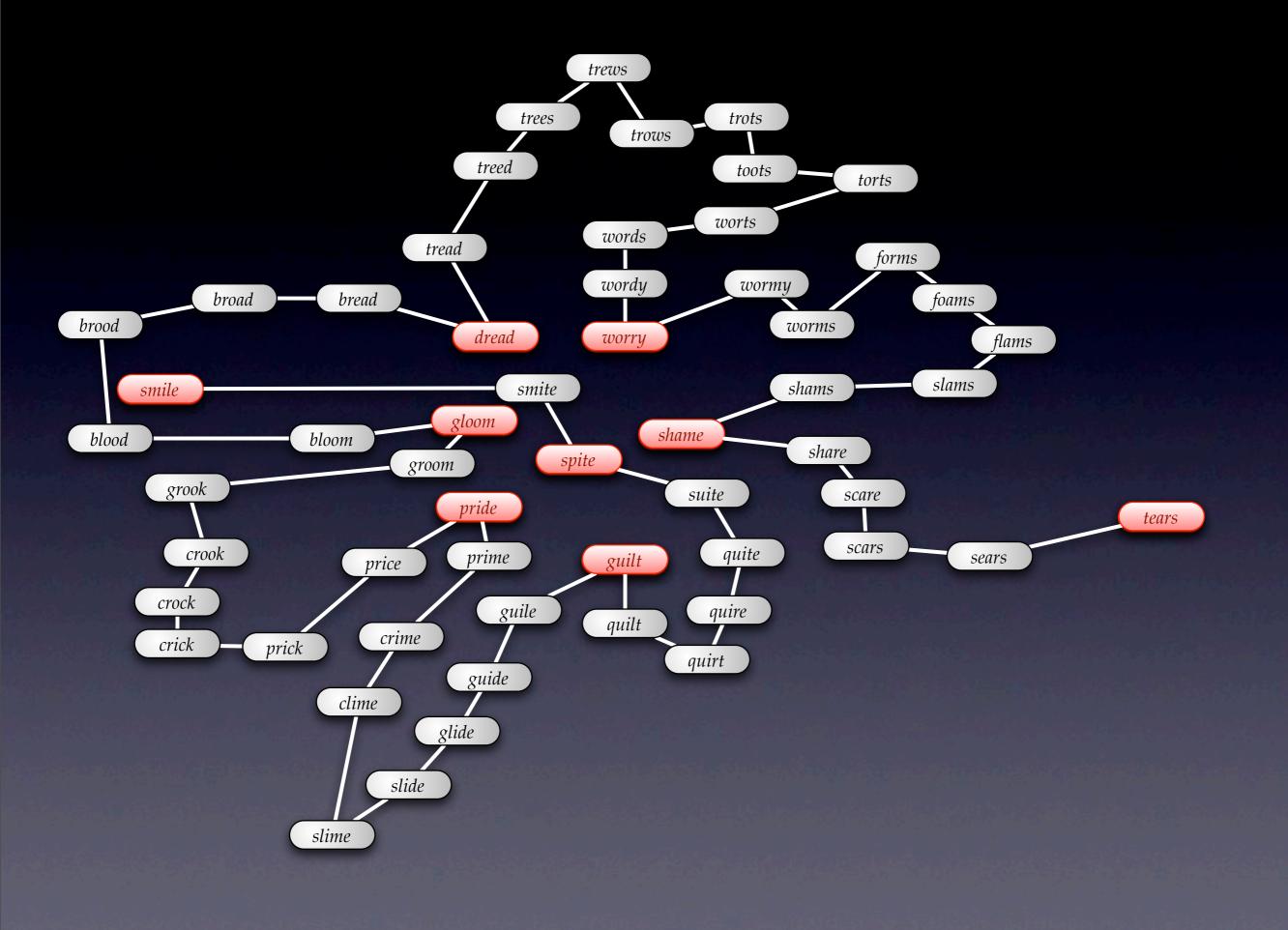
April 5.—Dip PEN into INK. Touch CHIN with NOSE. Change TEARS into SMILE.

268. c.

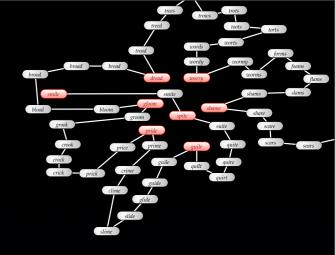
552.

Path through specified vertices

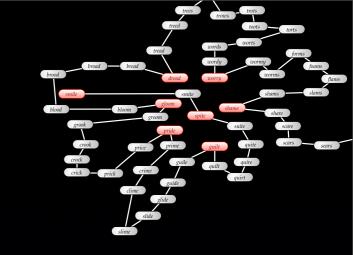




smusemodem^{nvoy}expatbosky bases easels awedjacks wackoposer filer bible medicredidbedimmidis pilot pulpy tally tampstabootabby fleas elope quite quilt polio polerpigmyfight orthonoted goner toner trues noseyhopespeweefermimergemetronourmolasmoles ruffs filmy miler fills fetid petit right fillet divesdomedrimer trues noseyhopespeweefermimergemetronourmolasmoles ruffs filmy miler fills fetid petit right fillet divesdomedrimer trues noseyhopespeweefermimergemetronourmolasmoles ruffs filmy miler fills fetid petit right fillet divesdomedrimer trues noseyhopespeweefermimergemetronourmolasmoles ruffs filmy miler fills fetid petit right fillet divesdomedrimer trues noseyhopespeweefermimergemetronourmolasmoles ruffs filmy miler fills fetid petit right filet divesdomedrimer trues noseyhopespeweefermimergemetronourmolasmoles ruffs filmy miler fills fetid petit right fillet divesdomedrimer trues noseyhopespeweefermimergemetronourmolasmoles ruffs filmy miler fills fetid vield visit avispy approximate avispy approximate avispy and pressive average and the construction of the state state st meanyheats beach reeks reify reeve fears teary quire quite quite array rajamadammaize aponwaspsbanal awns valid value paeandaily dills jello jelly parkaparts partypantylycra licks lippy tippy thaws the weather seedy shake pur spuppy ju sheen-up shoos shook want preve fears teary reeve fears teary quirequiting quirum quir fs enrol earth sates awky amas least be an bray stranks tage stave stain stays scalp scald scrap perks zetas speak skoal treap frost frosh krene urdscurly yelps keeps scour scops schmoselfs sportpunts cunts fluke pluckblush anise animal see unfedfame diamet amedialky sells knollknopsgooks ntra extra eaten sated gawksjapan refix boffs boffs boffs boffs boffs boffs boffs boffs boffs state state state state state state states balks bulksknowknowsnobs extolmateycanedraud gyppykepis biffy blimpflumesince suts silos sears hour parepageunazemusesmuter unty asker alibi colon billy bilk wispssissy noosecoonsy ogashapaxcasas obese tress tramp trite jails calls cabalrowannoral porche runs venal vigil nihil splat sulfa stagy slaps bobby bagel miles tilde ilty pixie fixes terrate erdyparedpawerpagesmakesnazednutesruntsabets glens cleft, wills mittes lisps prous afoul cloud clanga cornaportawash crash troll trees lades valet vagus radar comps winsbeefs deem seiny leans reads selah relax shagsstagshobby gator baldyhilumnixup fixit anvilad mittacky tacos bocci bocci nina guos shooner upper anter average and terminance gues chiefe wills mile tops buils information and upper available and the subscraft for the subscraft f dollydaddydazedjunta costa goest fingenappynuffsghostgamintacit sectsaxmanwokabenntagy climenvula drill trips snipsunitykoinelowedfork formsrings rinds cinchcheep theft swell dilly dinksdonorbozos bluff inure inert uteri questworstwurstglandstandspecs skews tiers litre ulcer man best admitted general test medicine server severe seve honedcynic chairorgangroutodouunduanset fusesmusosbigot lagerwager datered ucescudishads fordsforumnorthnormbongsbraves wankslangplasmplash plein penis peace reamsthong sheer osier onsel revueroams oud vaddy jaded gayergyves riled rated matesmires mixed coxes boned posessoapy stool stole culls mpdeweytenet lyres limemimedireasunmosflags flank blini acing agile twits tatty lotta cornudopeyropedkopek brier brief wooevanedsaxes taxed prateprankprowspoopsnookykookynooedboxespoxesvowedtdgedpy8myxges trial trice dicotcomfycoopscom infummy ubed turpy ummitimanipian aprovada ge clack belch belli balms tithe tutti lusty beret bodes coper impel freer freed alter alienwaxed faxes promoprome plops tooky roods routs vocab voxelvowerroars rearmerings attar vital visor cocos chops croft goofs go othsmotes betel hexer deter durst bts tares tune funnyturnsburntpurgeproveprivy knell idyls idyll decal darernakednukedtulip bulgebuoybrownbriar knelt hotly kooks poise notch fetch hengenedgevedgedozeddowelrewed sicko siege bleed gluer alley taper tipsy thaw/unapt unfit unzipuncle galls bulbs busts huskyhulkymutkyhuttywattswaled solve sulkymylamimer mer skier shier

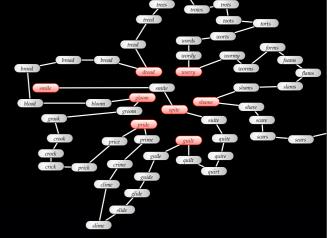


k=n: Hamilton path





k=n: Hamilton path no poly(*k*)-algorithm under P vs NP



k=n: Hamilton path no poly(*k*)-algorithm under P vs NP no exp(*o*(*k*))-algorithm under ETH

k=*n*: Hamilton path
no poly(*k*)-algorithm under P vs NP
no exp(*o*(*k*))-algorithm under ETH
Brute force: *O*(*n*!) (note: not *n^k*)

k=n: Hamilton path
no poly(*k*)-algorithm under P vs NP
no exp(*o*(*k*))-algorithm under ETH
Brute force: *O*(*n*!) (note: not *n^k*)
Disjoint paths: *f*(*k*)·poly(*n*)



Robertson–Seymour

k=n: Hamilton path
no poly(*k*)-algorithm under P vs NP
no exp(*o*(*k*))-algorithm under ETH
Brute force: *O*(*n*!) (note: not *n^k*)
Disjoint paths: *f*(*k*)·poly(*n*)
Algorithms for *k*=1,2,3.



Robertson–Seymour

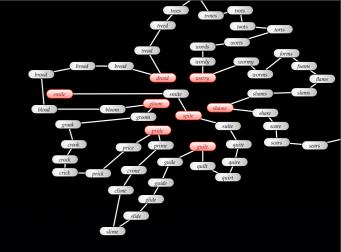
• *k*=*n*: Hamilton path • no poly(*k*)-algorithm under P vs NP no exp(o(k))-algorithm under ETH • Brute force: O(n!) (note: not n^k) • Disjoint paths: f(k)·poly(n) • Algorithms for k=1,2,3. • Algorithm in $exp(exp(k^{10}))$



Robertson–Seymour

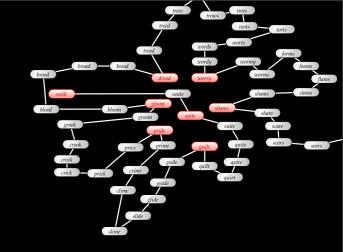


Kawarabayashi



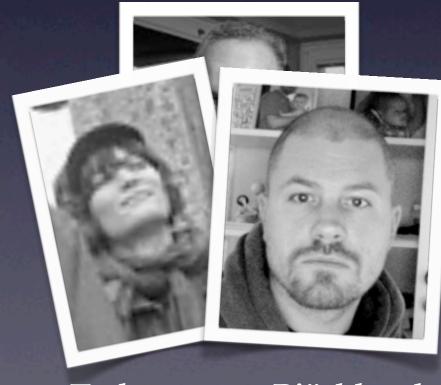
New result [SODA12]: randomised algorithm in time exp(k)poly(n)





New result [SODA12]: randomised algorithm in time exp(k)poly(n)

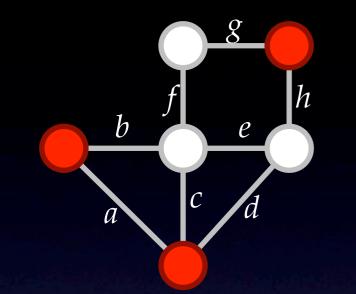
Thm: Shortest(!) cycle through k given vertices or edges in time 2^k poly(n) with exponentially small onesided error.



Taslaman

Björklund

Trick: Look at Polynomials Instead



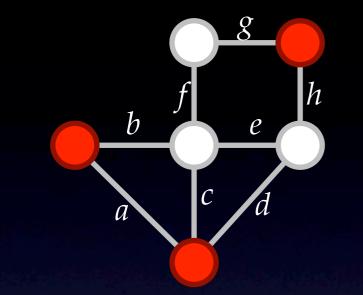
Koutis



Williams



Trick: Look at Polynomials Instead



Koutis



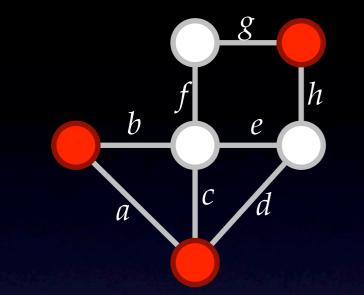
Williams



Björklund et al.



Trick: Look at Polynomials Instead



Koutis



Williams

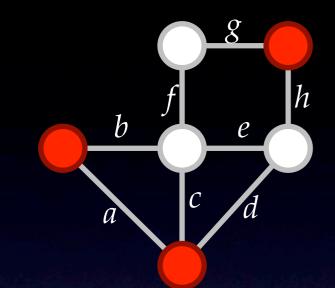


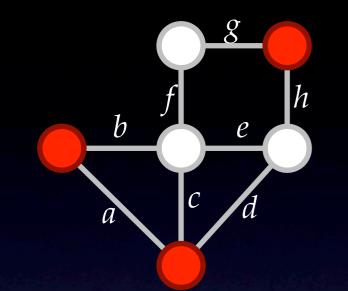
Björklund et al.



Tutte

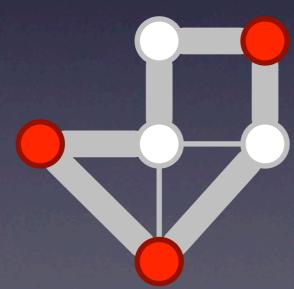


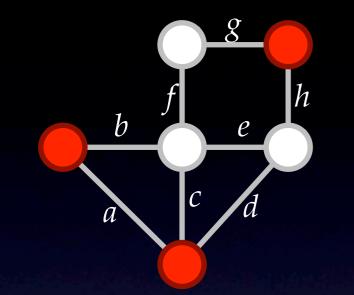




monomial for every walk

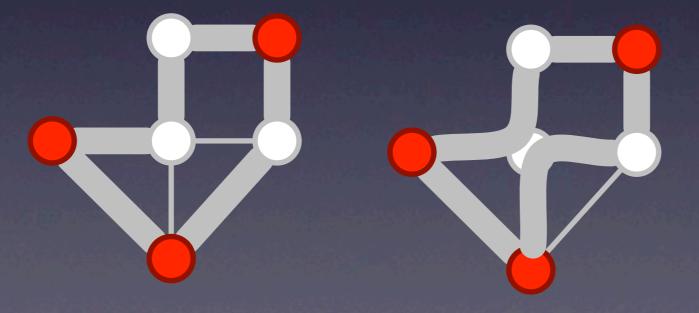
a·b·f·g·h·d

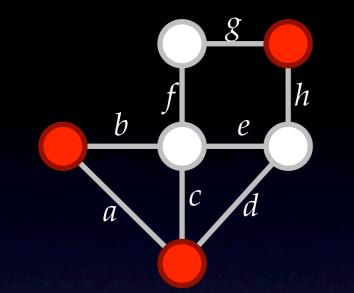




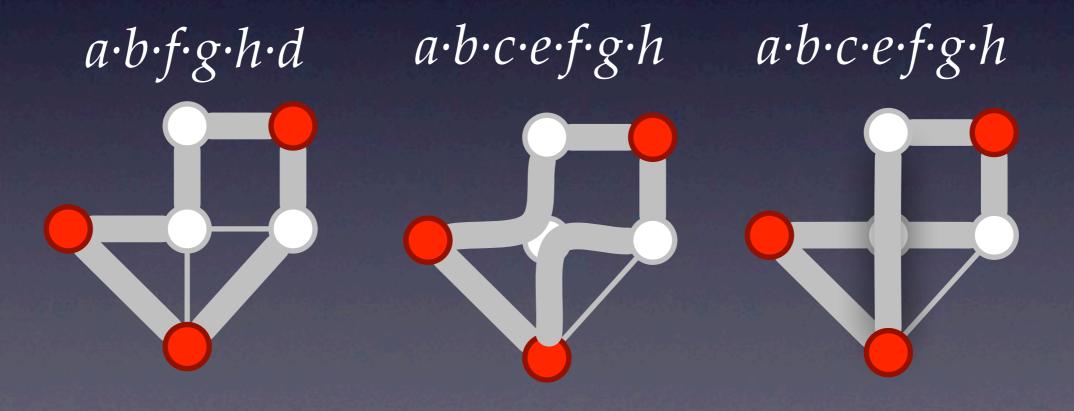
monomial for every walk sum over all walks

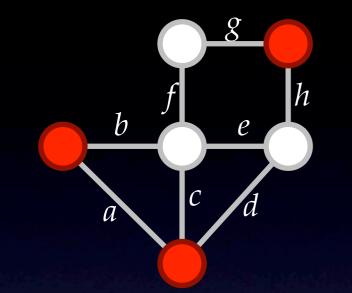
a·b·f·g·h·d a·b·c·e·f·g·h





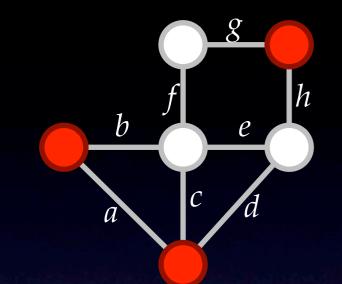
monomial for every walk sum over all walks





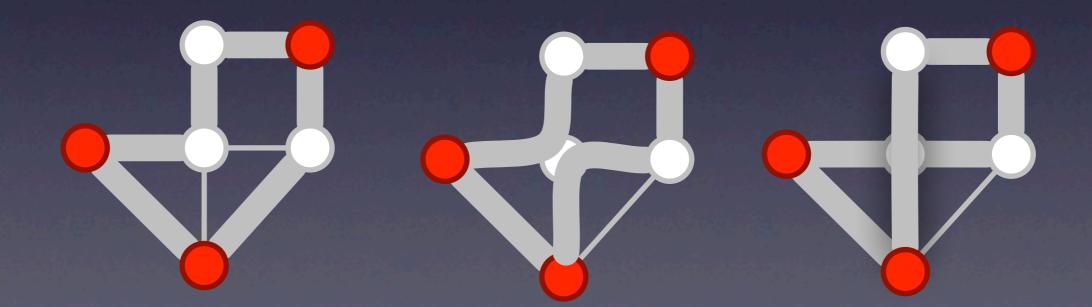
monomial for every walk sum over all walks mod 2

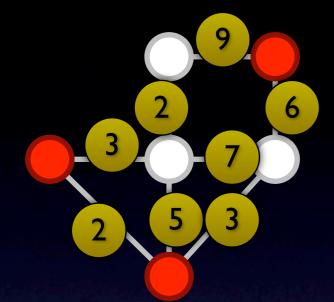




(Not really. Look at random numbers and *interpret* them as polynomial evaluations.)



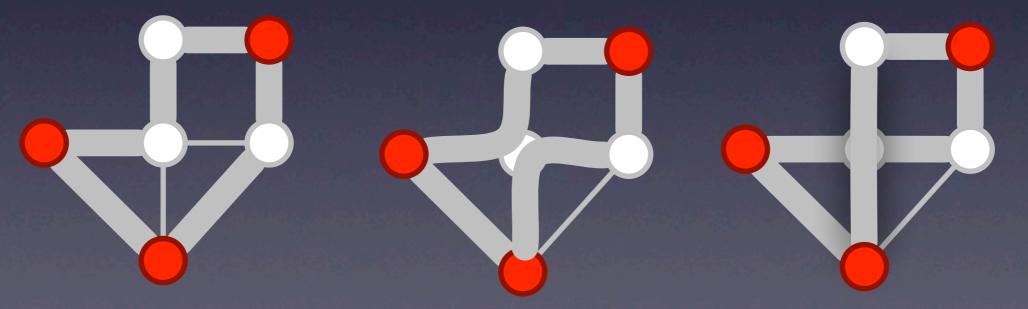




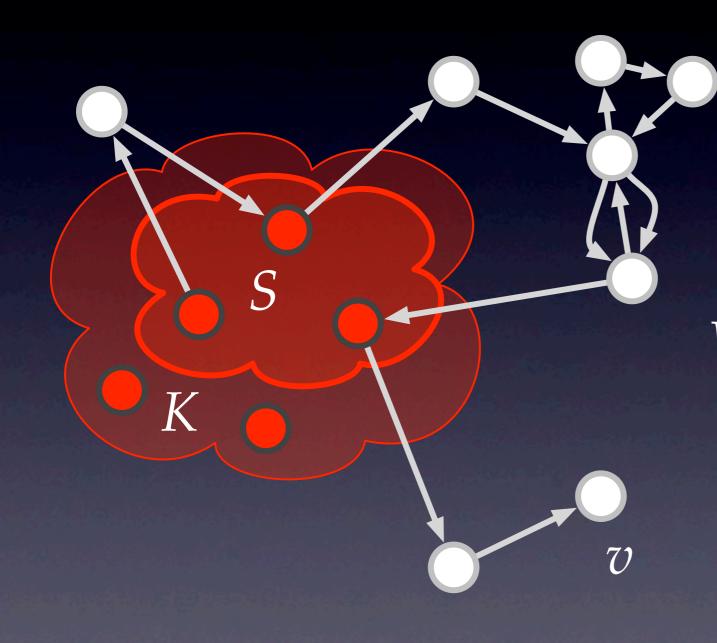
(Not really. Look at random numbers and *interpret* them as polynomial evaluations.)



$3 \cdot 2 \cdot 9 \cdot 6 \cdot 3 \cdot 2$ $3 \cdot 2 \cdot 9 \cdot 6 \cdot 7 \cdot 5 \cdot 2$ $3 \cdot 7 \cdot 6 \cdot 9 \cdot 2 \cdot 5 \cdot 2$



Constructing all Walks: Dynamic Programming for Sequencing Problems



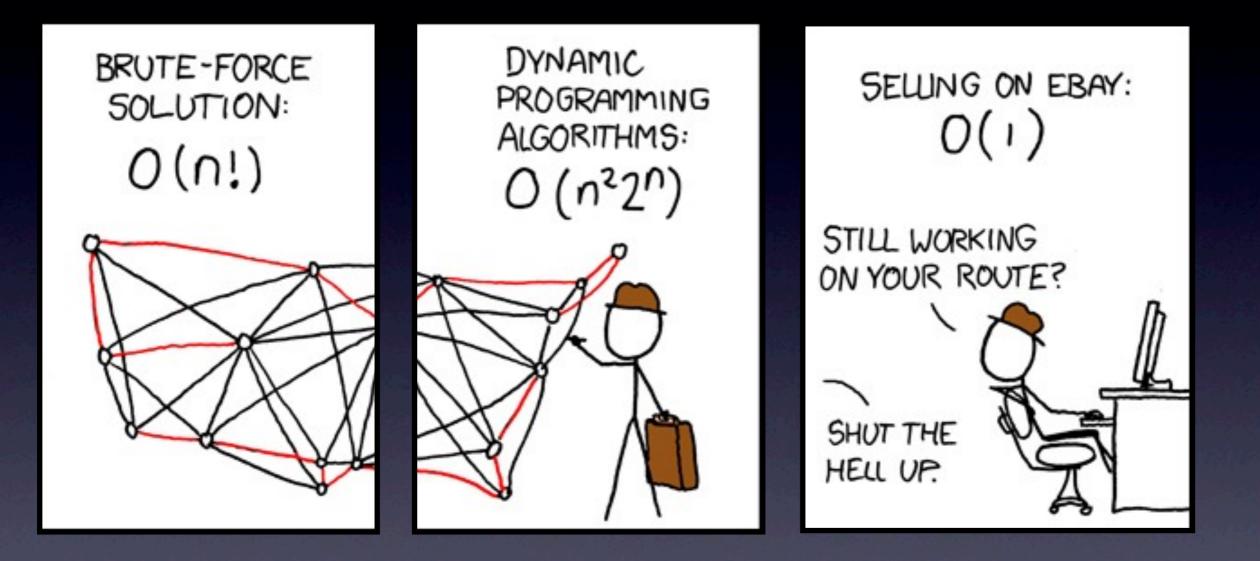
 $K = specifed \ vertices$ $S \subseteq K$

W(r,S,v) = walks s.t. length: r end in v visit all in S exactly once no other in K

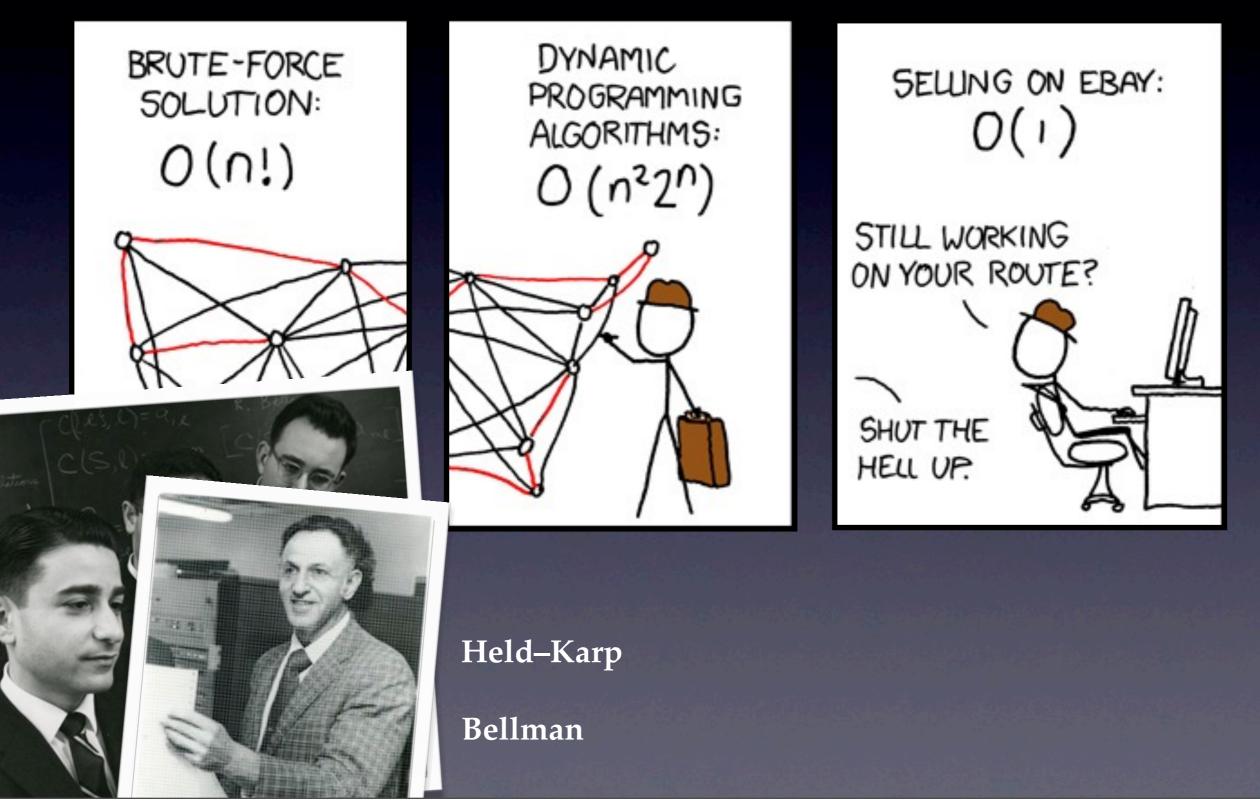
$$W(r,S,v) = \begin{cases} \bigcup_{uv \in E} W(r-1,S,u) & v \notin S \\ \bigcup_{uv \in E} W(r-1,S-v,u) & v \in S \end{cases}$$

Time: 2^Kpoly(n)

Constructing all Walks: Dynamic Programming for Sequencing Problems

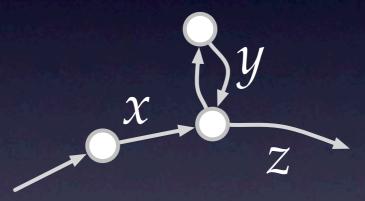


Constructing all Walks: Dynamic Programming for Sequencing Problems



onsdag 12 oktober 11



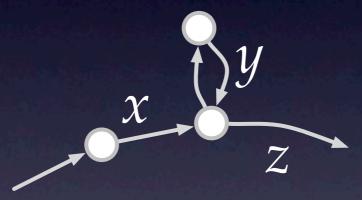


does not cancel



onsdag 12 oktober 11



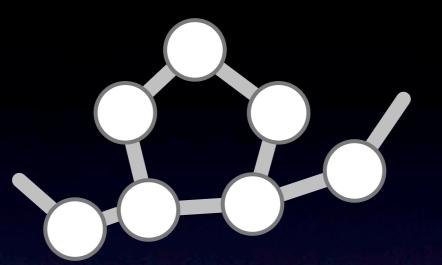


does not cancel

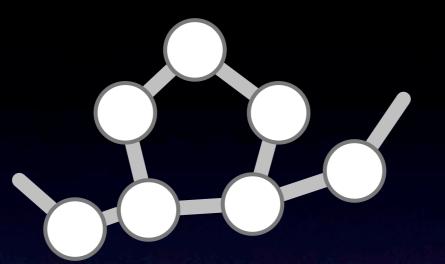


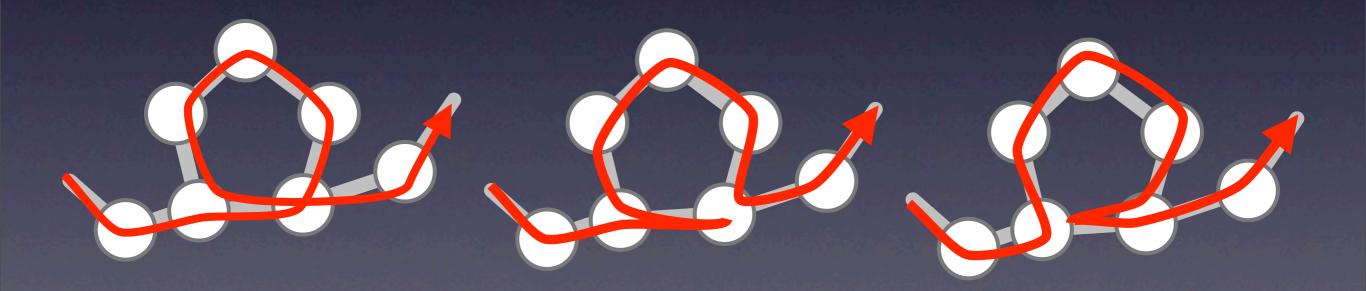
(solved in the dynamic program: just avoid "digons")



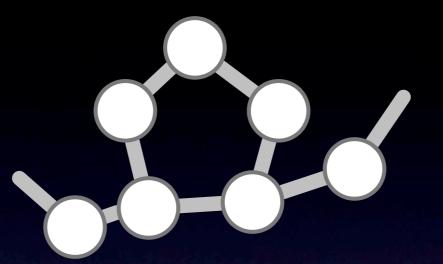








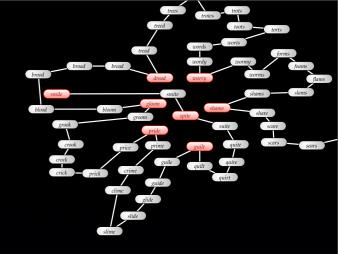






(solve shorter lengths first)

k specified vertices, n vertices

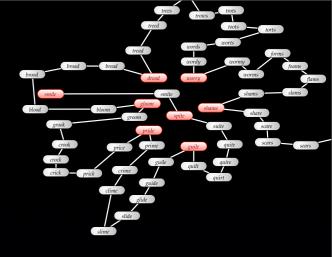


1. Associate random value from GF(2^{*q*}) to each edge

2. Use dynamic programming to count the contribution of all sufficiently well-behaved walks

3. Return "Found one!" if the result is nonzero

k specified vertices, n vertices

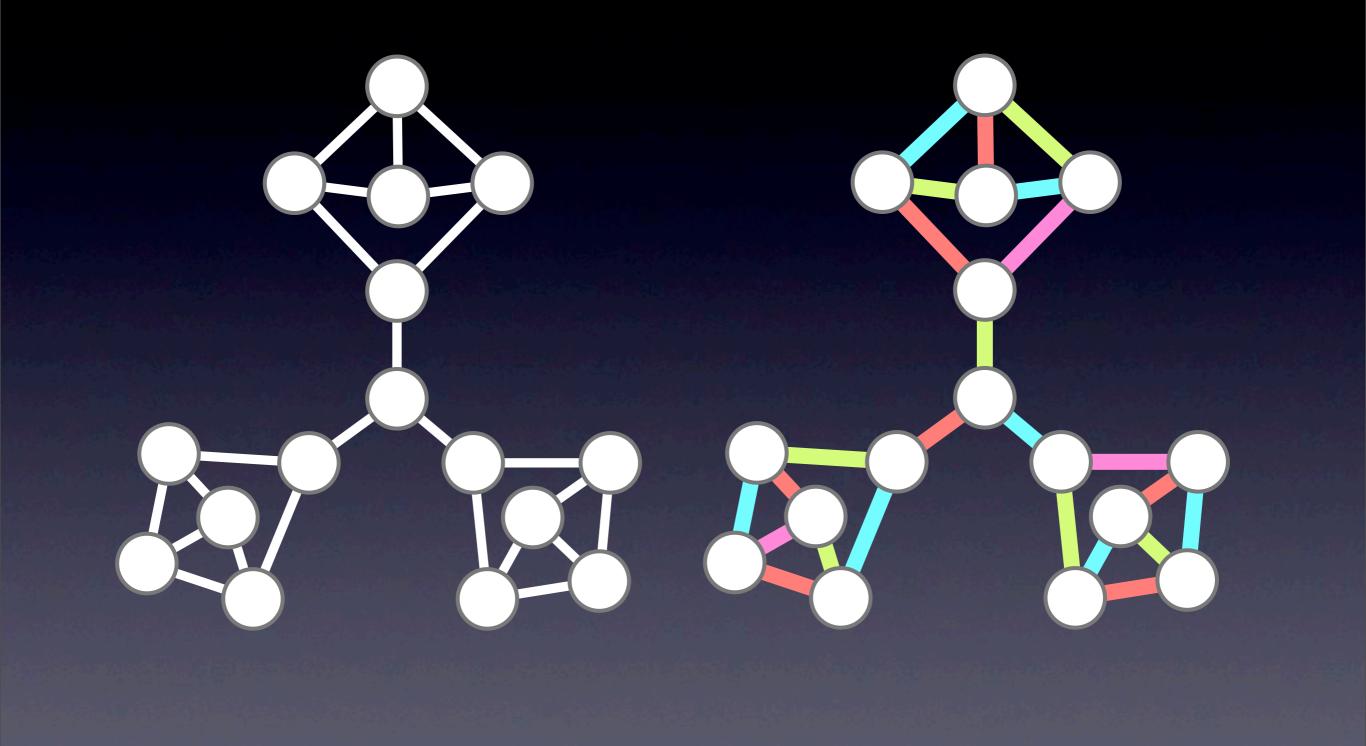


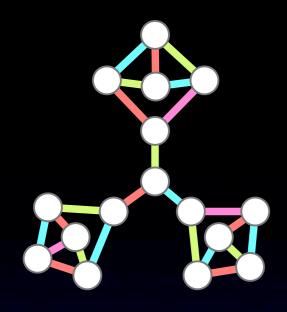
Theorem: Shortest cycle through k given vertices or edges in time 2^kpoly(n) with exponentially small one-sided error.

- 1. Associate random value from GF(2^q) to each edge
- 2. Use dynamic programming to count the contribution of all sufficiently well-behaved walks
- 3. Return "Found one!" if the result is nonzero

Edge Colouring

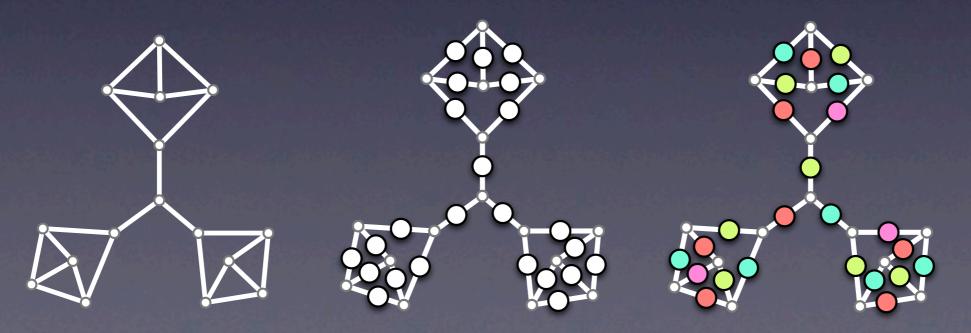


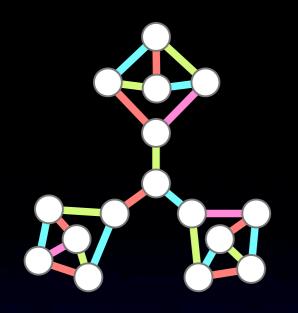




k: # colours d: degree Vizing: k = d or k = d+1

Brute force: check all d^m possibilities Vertex colour the line graph: time $2^m = 2^{nd/2}$





k: # colours d: degree

Brute force	d^m
Vertex colour the line graph	2 ^m =2 ^{nd/2}
"Narrow sieves" [BHKK]	$2^{n(d-1)/2}$

Under ETH: not in exp(o(n))



Edge Colouring takes $\begin{array}{c} exp(n) \\ d^n = exp(n \log d) \\ exp(m) = exp(nd) \end{array}$

onsdag 12 oktober 11

Tak fordi I kom