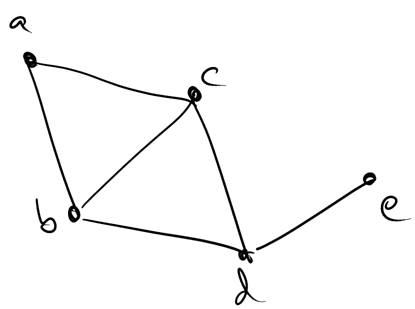
Math 343, lecture 19

① Cliques by backtracking

Def Let G be a graph. A clique of G is a

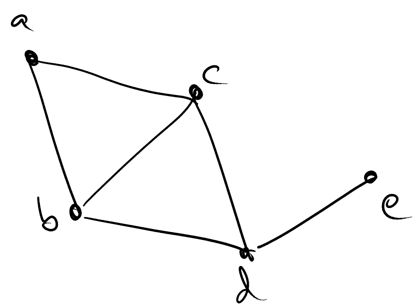
Note

eg $G =$ 

The cliques of G are:

Def Given a graph G and a clique S of G

S is **maximal** if

eg $G =$ 

maximal cliques

Note

We would like to be able to

The first thing we need to do is

The partial solutions will be

We can grow a partial solution by

How do we do that efficiently and without repetition?

Algorithm AllCliques

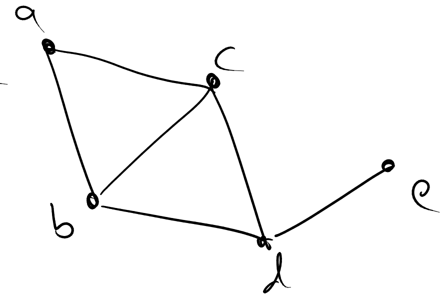
global X, C_ℓ, N_ℓ, $\ell = 0, 1, \ldots, n-1$

input ℓ

if $\ell = 0$

else

eg $G =$



if $\ell = 0$

then ϕ is a clique

N_ℓ = the set of all vertices of G

C_ℓ = the set of all vertices of G

else

$\{x_0, .., x_{\ell-1}\}$ is a clique

$N_\ell = A_{x_{\ell-1}} \cap N_{\ell-1}$

$C_\ell = A_{x_{\ell-1}} \cap B_{x_{\ell-1}} \cap C_{\ell-1}$

if $N_\ell = \phi$

$\{x_0, .., x_{\ell-1}\}$ is a maximal clique

for $x_\ell \in C_\ell$

AllCliques$(\ell+1)$

② Runtime

The size of the tree is key to the runtime.

Let $c(G) =$

Then

What is $c(G)$?

worst case is

What about average case?

What is the average value of $c(G)$?

To be precise say we run over all graphs on n (labelled) vertices

call this $\mathcal{G}_n$

note $|\mathcal{G}_n| =$

let $\bar{c}(n) =$

What is this sum?

with a bit of work can show it is $O(n^{\log_2 n + 1})$

so the algorithm has run time $O(n^{\log_2 n + 2})$

this is quasipolynomial time

③ Next time

Smarter pruning by bounding