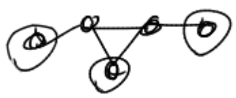
CS 466 Fall 2013 — Lecture 23 - fixed parameter tractable

Anna Lubiw
Univ. Waterloo

Recall: Fixed Parameter Tractable Algorithm (FPT)
has run time $O(f(k)\, n^c)$

- n - input size
- k - a parameter of the input.
- $f(k)$ - function of k, independent of n
- c — constanct, independent of k.

Ex. Ind. Set - Given a graph G, does it have an ind. set $\geq k$?



Ind. set of size 3

Brute force $O(\underbrace{n^k}_{\#\ k \text{ subsets of } n \text{ vertices}} \cdot (n+m))$

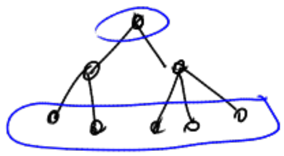
NOT Fixed parameter tractable.

FACT: No one has found a FPT alg. for Ind. Set (for this parameter).
(Note: would such an FPT alg imply P = NP? No one knows.
Note: corrected from last day)

Ex. Vertex Cover parameter = size of min V.C.
FPT alg. $O(2^k \cdot n)$ — using branching alg.
$O(2^k \cdot k^2 + m + n)$ — kernelization.

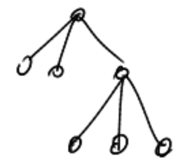
Today: Ind. Set and FPT with a parameter that measures how "tree-like" is our graph.

Ind. Set on a tree



? start at leaves?

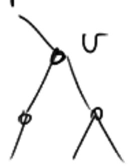
can't just take levels



– with weights on vertices, $w(v)$

Dynamic Programming

Subproblems



$IS(v)$ – max weight of ind. set in subtree rooted at v

$IS^{\circ}(v)$ – " " " " that DOES NOT include v

Alg. QUIZ.

Initialize: for every leaf v

$$IS^{\circ}(v) = \boxed{0}$$

$$IS(v) = \boxed{w(v)}$$

For all nodes v in leaf-root order

v has children $u_1 \cdots u_t$

$$IS^{\circ}(v) = \sum_{i=1}^{t} IS(u_i)$$

$$IS(v) = \max\left\{ w(v) + \sum_{i=1}^{t} IS^{\circ}(u_i),\ IS^{\circ}(v) \right\}$$

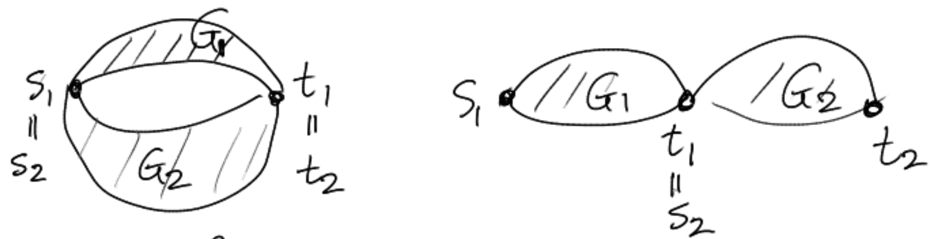
Return $IS(\text{root})$

Can use same idea for graphs that are "close to" trees.

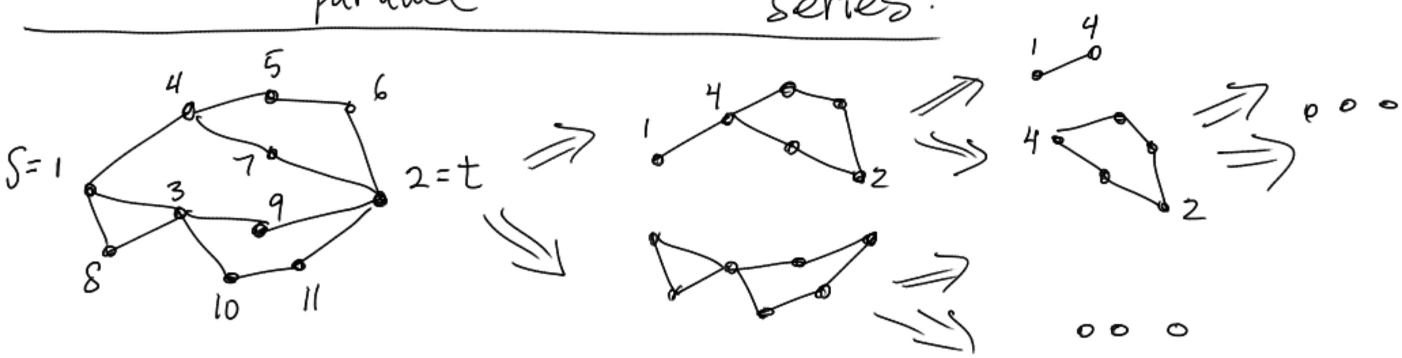
Series-parallel graphs (SP graphs)
defined recursively — graph + two "terminal" nodes s, t.

- 

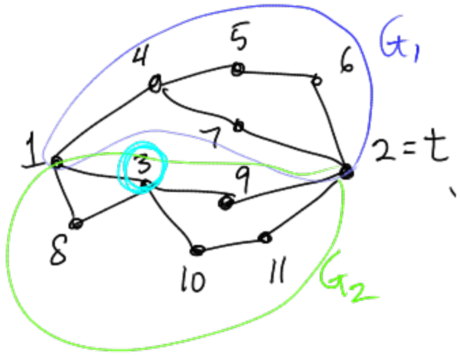
is SP

- 

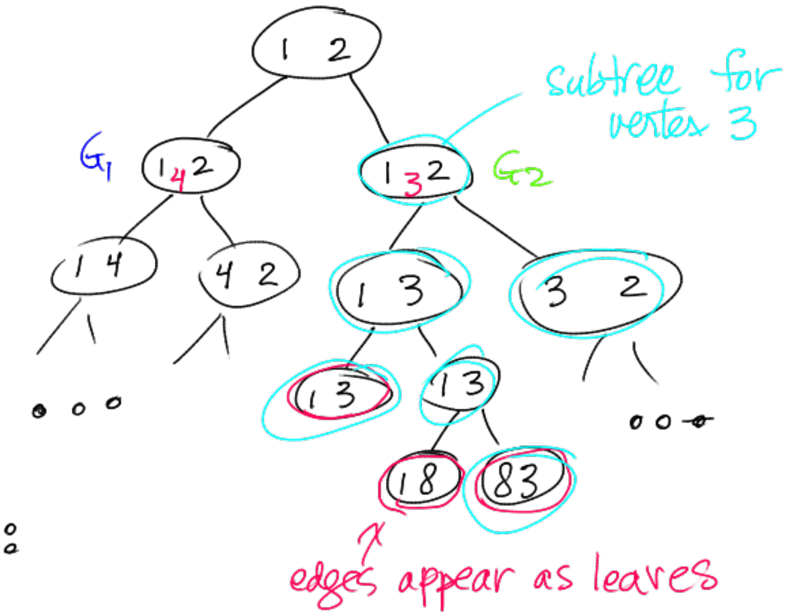
parallel series.



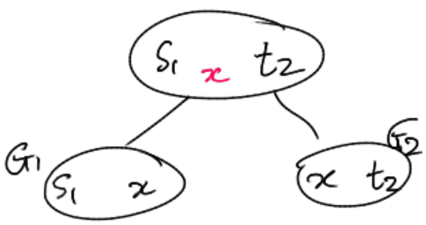
Ind. Set in series parallel graph.
We can do dyn. programming based on
- max. ind. set including s and t
- - . . . . . s, not t
- - not s, t
- - - not s, not t.

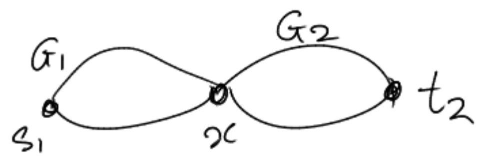


We can model as a tree



For each "series" step, include middle terminal in parent node of tree:



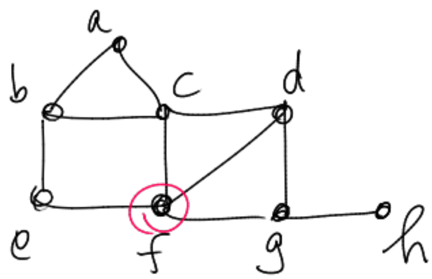


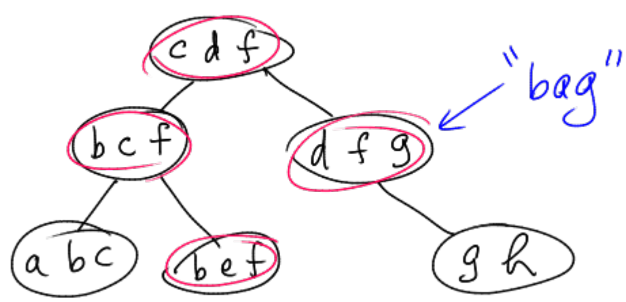
Properties

1. If $e=(u,v)$ is an edge of G then u and v appear together in a tree node
2. Every vertex v of G corresponds to a subtree of T

Generalization Tree-width. – Robertson & Seymour Graph Minors Project 20 papers, 500 pages.

represent graph as a tree





We have properties 1 & 2

width of decomposition = size of largest bag − 1

<u>tree-width of G</u> − min. width of any tree decomposition. $\leq n-1$ (put all vertices in one bag)

and we only need $\leq n$ bags in any tree decomposition.
(this is not hard to prove).

Graphs of tree-width 1 = forests (subgraphs of trees)

- - - - 2 = subgraphs of series-parallel

FACT: Finding tree-width of a graph is NP-hard.

But there is an FPT alg.

$O(2^{O(k^3)} \cdot n)$

Thm Max Weight Ind. Set in graph of tree-width k can be found in time $O(2^k \cdot n)$

Idea: use dynamic programming, working up the tree.

For each bag B (size $\le k+1$)

we find, for each subset $A \subseteq B$ (there are $O(2^k)$ of them)

a max weight ind. set including A, excluding $B-A$.

in subtree rooted at B.

Other problems FPT in tree-width

- 3-colouring
- min. colouring
- Hamiltonian cycle (more complicated - still dyn. prog.)

Hardness results.

all relative results of form:

FPT alg. for problem X $\Rightarrow$ FPT alg. for problem Y

proved via reduction that preserves parameter & FPT.

(technical).