21. Bounded Search Trees

Observation: Source of high running times is branching behavior of algorithms

Idea: Keep this branching a function of the parameter.

Approach:
- First, compute some search space, often a search tree, of size bounded by a function of the parameter (typically exponential).
- Then, run some relatively efficient algorithm on each branch of the tree.

Background:
- Exponential worst-case complexity stems from problem instances where a complete exploration of the search tree is required.
- Experiments may show that often only a small portion of the data set is explored.
- Parameterized complexity tries to close this gap.

Theorem (Monien, Downey & Fellows):

VERTEXCOVER is solvable in time $O(2^k \cdot |V(G)|)$.

Proof:

We construct a binary tree of height k as follows.

Label the root by $(\emptyset, G)$.

Choose an edge $\{v, w\} \in E$.

In any vertex cover V' of G, we must have $v \in V'$ or $w \in V'$.

So we create two children of the root node,

one labelled by $(\{v\}, G \setminus \{v\})$ // Remove v and all edges incident to it.

the other labelled by $(\{w\}, G \setminus \{w\})$.

Then we continue from (S_1, G_1) in the same way (pick an edge).

↳ The set S labelling a node represents a potential vertex cover, the graph shows what remains to be covered

↳ If the graph has a vertex cover $\le k$, then we find $(S, \emptyset)$ in a tree of height $\le k$. □

21.1 Shrinking the Search Tree

Goal: Improve the constants.

Idea: Understand and exploit the combinatorial structure of input instances.

This is important! Everywhere in algorithmics!

Argumentation: Fix a k and consider graph G.

- If G only has vertices of degree ≤ 2, it consists of
  - paths and
  - cycles and
  - isolated vertices.
- If such a G has $> 2k$ edges it cannot have a size $\leq k$ vertex cover (every vertex kills two edges, but there are more than $2k$ edges to kill).

$\Rightarrow$ Wlog. we can study graphs with vertices of degree ≥ 3 (wlog. will cost us a poly(k) per branch, eaten by O).

- Choose a vertex of degree ≥ 3, say v_0. Either v_0 is in a vertex cover or all its neighbors are.
- Start a search tree with
  - ↳ one branch labelled $\{v_0\}$
  - ↳ the other branch labelled $\{w_1, \dots, w_p\}$, $p \geq 3$.

  Again consider the subgraphs:

  G_1 not covered by $\{v_0\}$ and

  G_2 not covered by $\{w_1, \dots, w_p\}$.

In G_1, need a vertex cover of size $k-1$.

In G_2, need a vertex cover of size $k-p$.

Again, we only consider subgraphs with degree ≥ 3 nodes.

· The complexity is again determined by a tree of size $O(\underbrace{2^k})$ number of leaves, size was $2^{k+1}-1$.
but it is now smaller.

The recurrence that gives the number of <u>leaves</u> depending on k is:

$$\left.\begin{array}{ll} a_0 := 1 & // \ k=0 \\ a_1 := 1 & // \ k=1 \\ a_2 := 1 & // \ k=2 \end{array}\right\} \text{Input graphs are not split up.}$$

$$a_3 := \underbrace{a_2}_{\substack{k-1 \text{ cover} \\ \text{needed}}} + \underbrace{a_0}_{\substack{k-p \text{ cover} \\ \text{needed.}}}$$

In general:

$$a_k := a_{k-1} + a_{k-p} \, .$$

There is a standard technique for bounding such functions asymptotically.
Prove by induction on k that $a_k \leq c^k$ for some $c > 1$
but as small as possible.

What values of c are good? Need

$$c^k \geq c^{k-1} + c^{k-p}$$

$$c^p \geq c^2 + c^0 .$$

The latter leads to the so-called <u>characteristic equation</u>:

$$c^p - c^2 - 1 = 0$$

<u>Solve</u> the characteristic equation.

<u>Note</u>: There is not always a unique positive solution.

In our setting when $p = 3$:

$$c^3 - c^2 - 1 = 0.$$

Take $c = 5^{\frac{1}{4}}$. Then

$$c^2 + 1 = \sqrt{5} + 1 \leq 5^{3/4} = c^3.$$

With this, $a_k \leq 5^{\frac{k}{4}}$
can be verified by induction.

<u>Theorem (Balasubramanian, Fellows, and Raman)</u>:
VERTEXCOVER can be solved in $\mathcal{O}((5^{\frac{1}{4}})^k \cdot |G|)$.

Note that $5^{\frac{1}{4}} \leq 1{,}466.$

There are better bounds based on a more refined combinatorial analysis of the neighborhood.