DELFT UNIVERSITY OF TECHNOLOGY
Faculty of Electrical Engineering, Mathematics and Computer Science

Exam IN4344 Advanced Algorithms – Part II

November 7, 2022 | 9:00–12:00

- This is a closed-book individual examination with 6 questions worth 50 points in total.
- If your score is n points, then your grade for this exam part will be $1 + \frac{9}{50}n$.
- Use of (graphical) calculators is not permitted.
- Write clearly, use correct English, and avoid verbose explanations. Giving irrelevant information may lead to a reduction in your score. *Almost all question parts can be answered in a few lines!*
- This exam covers Chapters 10 of Kleinberg, J. and Tardos, E. (2005), *Algorithm Design*, all information on the slides of the Part II of the course, everything discussed in lectures of Part II, and the supplemental study material provided via BrightSpace.
- The total number of pages of this exam is 5 (excluding this front page).
- Exam prepared by M.M. de Weerdt. (With thanks to Leo van Iersel.)

1. (8 points) The *vaccination problem* has as input parameters d and k both in $\mathbb{N}$, and an undirected graph $G = (V, E)$. The output of the decision problem is YES iff a set $C \subseteq V$ exists with $|C| \leq k$ such that for each $v \in V \setminus C$ holds that at most d neighbors of v are not in C.

   The interpretation of the problem is that the vertices of the graph G represent persons and its edges represent contacts (potential virus transmissions). The goal is to vaccinate at most k people in such a way that each person that is not vaccinated has at most d contacts that are also not vaccinated (to limit the speed the virus spreads).

   Describe a bounded search tree algorithm for this problem (include bases cases and explain why the answers to the subproblems provide an answer to the parent problem). Provide and explain an analysis of the runtime and explain why this problem is FPT.

   Solution:

   1. Base case NO: if $k = 0$ and there is a vertex with degree at least $d + 1$.
   2. Base case YES: if all vertices have degree at most d (and $k \geq 0$).
   3. Observe that if $k > 0$ and there is a vertex v of degree $d + 1$ or more, one of the first $d+1$ neighbors, or v must be in C, because we cannot let v have $d+1$ neighbors. In this case branch into $d + 2$ subproblems:
      (a) create the branches as follows: put v in C, put a neighbor for each $u_i \in \{u_1, u_2, \ldots, u_d, u_{d+1}\}$ of v in C
      (b) remove this vertex from the problem and reduce k by one
   4. The depth of the search tree is at most k (and each node branches into at most $d + 2$ subproblems).
   5. The runtime thus is $O^*((d + 2)^k)$.
   6. This problem is fixed-parameter tractable because its runtime is exponential only in the parameters.

2. Consider the following problem called Colourful Path, where we look for paths that contain all colours exactly once.

   Let $C = \{1, 2, \ldots, k\}$ be a set of k colours, $G = (V, E)$ an undirected graph and $f : V \to C$ a function that assigns a colour $f(v) \in C$ to each vertex in V. Let all of these be given to us. For $S \subseteq C$, define an S-path to be a path P such that it contains for each colour $i \in S$ exactly one vertex $v \in P$ with $f(v) = i$. The decision problem we consider is: does G contain a C-path?

   (a) (1 point) If a C-path exists, what is its length?

   Solution: k vertices, or $k - 1$ edges; both are correct answers

   (b) (3 points) Let $S \subseteq C$ and $v \in V$ be given. Prove that there exists an S-path starting at v if and only if there exists a $S \setminus \{f(v)\}$-path starting at a neighbour of v.

Solution: We consider two directions. First suppose there exists an S-path starting at v. Removing v from the path gives an $S \setminus \{f(v)\}$-path starting at a neighbour of v.
Now suppose there exists an $S \setminus \{f(v)\}$-path starting at a neighbour of v. Then this path doesn't use v since it only uses vertices with colours in $S \setminus \{f(v)\}$. So we can add v to this path and obtain an S-path starting at v.

(c) (4 points) Give a recursive dynamic programming function for Colourful Path with a runtime not exponential in the size of the graph. (Hint: do not forget the base case and how you would use the function to find the solution.)

Solution: For $v \in V$ and $S \subseteq C$, define $\mathsf{OPT}(S, v) = 1$ if there is an S-path starting at v and $\mathsf{OPT}(S, v) = 0$ otherwise.

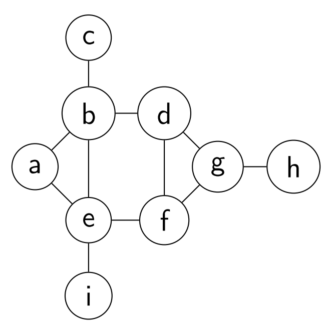
- Base case For $|S| = 1$: $\mathsf{OPT}(\{i\}, v) = 1$ if $f(v) = i$ and $\mathsf{OPT}(\{i\}, v) = 0$ otherwise.
- For $|S| > 1$: $\mathsf{OPT}(S, v) = \max_{u \in N(v)} \mathsf{OPT}(S \setminus \{f(v)\}, u)$ if $f(v) \in S$ and 0 otherwise.
- The answer is yes if and only if $\max_{v \in V} \mathsf{OPT}(C, v) = 1$.

Correctness follows by part (a).

(d) (1 point) Determine the runtime of your algorithm.

Solution: For each vertex $v \in V$ and for each subset of C, we loop through all (at most $|V|-1$) neighbours of v. So the runtime is $O(2^k |V|^2)$. $O^*(2^k)$ also gets full points. ($O^*(n^k)$ or $O^*(d(G)^k)$ where $d(G)$ is the degree of G do not get points, because then the point of benefit of the proposed dynamic program has been missed.)

3. (8 points) Give a tree decomposition of the following graph that has the lowest width you can find, *and* explain why this is a correct tree decomposition (hint: you don't need to give the definition itself, but you may use it for your explanation). (There will be $|w' - w|$ point deduction if the width of your tree decomposition is w' while the treewidth is w.)



Solution: The answer maybe given by a drawing of a tree – as long as it is clear which vertices from the graph are in the bag of which tree nodes, or the answer may be given in set notation – where both the tree structure and the bags need to be defined: $T = (V, E)$ with $V = \{1, 2, 3, 4, 5, 6, 7\}$,
$E = \{(1, 2), (2, 3), (3, 4), (4, 5), (4, 6), (6, 7)\}$ and the bags $\{cb\}, \{abe\}, \{bde\}, \{edf\}, \{ei\}, \{dfg\}, \{gh\}$.

Explanation: this tree meets the three other properties of a tree decomposition:

- every vertex is represented in one of the bags of the tree nodes
- for every edge, there is a tree node with both end points of that edge
- for every vertex, the bags of tree nodes containing this vertex is a connected subtree

4. Let $G = (V, E)$ be a clique.
   (a) (2 points) What is the treewidth $tw(G)$ of G?

   Solution: This is $|V| - 1$.

   (b) (6 points) Give a proof of your answer.

   Solution: (from slide 12)

   1. Take an arbitrary tree decomposition Tr for G with root node t .
   2. For every $v \in V$, find the node t_v closest to t such that $v \in V_{t_v}$.
   3. Then take $w \in V$ associated with tree node t_w with maximum distance from t .
   4. Take any $v \in V$. Since G is a clique, edge (w, v) needs to be covered in Tr, and since w only occurs in the subtree of t_w, with coherence vertex v should be in V_{t_w} .
   5. This holds for every $v \in V$, so $V \subseteq V_{t_w}$; thus Tr has width of at least $|V| - 1$.
   6. This holds for every tree decomposition and $tw(G) \leq |V| - 1$, so $tw(G) = |V| - 1$.

   Other formulations are possible that do not follow the above steps exactly

5. (7 points) Consider the well-known problem of finding a maximum-weight independent set in an undirected graph $G = (V, E)$ with for every vertex $i \in V$ a weight w_i.

   Consider the exact decision diagram constructed using some ordering of vertices $i \in V$ and the following recursive function:

$$\mathrm{OPT}_i(S) = \begin{cases} \max\{\mathrm{OPT}_{i+1}(S \setminus \{i\}), \mathrm{OPT}_{i+1}(S \setminus N^*(i)) + w_i\} & \text{if } i \in S \\ \mathrm{OPT}_{i+1}(S) & \text{otherwise} \end{cases}$$

   Where $N^*(i)$ denotes the set of i and its neighbors. Additionally, the base case $\mathrm{OPT}_i(\emptyset) = 0$ for every $i \in V$.

Give a possible merge operator $\oplus$ to construct a relaxed diagram with smaller width, explain how this is used and why/when this is a relaxation.

Solution: Consider the following merge operator:

$$\oplus(S_1, S_2) = S_1 \cup S_2$$

at any level i where we aim to reduce the width, for any two nodes with similar subsets.

This is a relaxation, because every sequence/path from root to sink in the original decision diagram is also a valid path here, since the argument S explicitly represents the vertices which may be included in the independent set, but also new possible paths (and thus solutions) may be introduced.

6. Consider the following parameterized problem of Set Splitting. We are given a finite set U (the "universe"), a set $\mathcal{F}$ of subsets of U and an integer k. Can we colour the elements of U with two colours such that at least k sets of $\mathcal{F}$ are bi-chromatic (i.e. contain vertices of both colours)?

   (a) (2 points) Consider the following reduction rule. If $S \in \mathcal{F}$ is size 1, then delete S from $\mathcal{F}$. The parameter k remains the same. Show that the reduced instance has a solution *if and only if* the original instance has a solution.

   Solution: Suppose $(\mathcal{F} \setminus \{S\}, k)$ is a yes-instance. Then $(\mathcal{F}, k)$ is also a yes instance using the same colouring.
   Now suppose $(\mathcal{F}, k)$ is a yes-instance. Observe that S cannot be bi-chromatic because it has only 1 element. Hence, $(\mathcal{F} \setminus \{S\}, k)$ is also a yes-instance, again using the same colouring. (Alternatively, as always when proving A implies B, it is also possible – by contraposition – to prove that not B implies not A.)

   (b) (4 points) Prove that, if the condition of the previous reduction rule does not apply, and there exists an $x \in U$ that is in at least k sets in $\mathcal{F}$, then the answer is YES.

   Solution: Use one of the two colours only to colour x and the other one to colour $U \setminus \{x\}$.

   Since the previous reduction rule does not apply, each set has size at least 2. Hence, each of the at least k sets containing x are bi-chromatic.

   (c) (4 points) Consider the following reduction rule. If $S \in \mathcal{F}$ has $|S| \geq 2k$ then delete S from $\mathcal{F}$ and reduce k by 1. Show that the reduced instance has a solution *if and only if* the original instance has a solution.

   Solution: Suppose $(\mathcal{F}, k)$ is a yes-instance. Then clearly $(\mathcal{F} \backslash \mathcal{S}, k-1)$ is also a yes-instance using the same colouring.
   Now assume $(\mathcal{F} \setminus \{S\}, k-1)$ is a yes-instance. Then there exist $k-1$ sets from $\mathcal{F} \setminus \{S\}$ that are bi-chromatic. Select two differently-coloured elements from each of these $k-1$ sets. We have then selected at most $2k-2$ elements. Since $|S| \geq 2k$, we know that S

contains at least two elements that have not been selected. If we change the colour of these elements such that they have different colours, then S is bichromatic and each of the $k-1$ sets is still bichromatic. Hence $(\mathcal{F}, k)$ is also a yes-instance.