Institut for Matematik og Datalogi
Syddansk Universitet

May 2, 2023
JBJ

DM553/MM850 – Spring 2023 – Weekly Note 14

Stuff covered in Week 18

- Adversary lower bound for comparison based sorting. My notes Section 7. (Video 24)
- Information theoretical lower bound for comparison based sorting. Baase Section 2.4. (Video 24)
- Fixed parameter tractability from Cygan et al Chapters 1 and 2. (Video 25)
- Introduction to exact algorithms. Based on pages 1-6 from the book by Fomin and Kratz as well as pages 51-55 from the book M. Cygan et al, Parametrized Algorithms, Springer. You can find these pages on the homepage of the course as well as on the plan for this week in Itslearning. (Video 26)

Key points

- The information theoretic argument for the lower bound for comparison based sorting is unsatisfactory in the sense that it does not really give a clue how one, as an adversary, could force any given sorting algorithm to use $cn \log n$ comparisons on some input. In my notes I describe how to perform a strategy (answering queries of the form "is $x < y$?") in such a way that every comparison based sorting algorithm must use $\frac{n}{2} \log n$ comparisons in the worst case: for a given comparison based sorting algorithm $\mathcal{A}$ the adversary constructs an input (which is a permutation of $\{1, 2, \ldots, n\}$ and which depends on $\mathcal{A}$) that forces $\mathcal{A}$ to use at least $\frac{n}{2} \log n$ comparisons before obtaining the correct (sorted) sequence. We also saw that the adversary can perform this strategy very efficiently.
- Suppose we have a problem Q and some parameter k related to Q. We say that Q is in the complexity class FPT with respect to the parameter k if there is an algorithm $\mathcal{A}_Q$ which solves an instance $< I, k >$ of Q in time $O(f(k)n^c)$ for some computable function f and some constant c.
- A **Kernel** for an instance $< I, k >$ of a parameterized problem Q (with parameter k) is another instance $< I', k' >$ such that $|I'| + |k'| \leq g_Q(k)$ for some function g_Q (depending on Q), with the property that $< I, k >$ is a yes-instance of Q if and only if $< I', k' >$ is a yes-instance of Q (where, as usual, $< I, k >$ is a yes-instance of Q if $< I, k > \in Q$). We usually want a **kernelization algorithm** for Q which given an

instance $< I, k >$ computes a kernel $< I', k' >$ in time $O(f(k))$ for some computable function f.

- I proved how to obtain two different kernels for the vertex cover problem. The first one is obtained using reduction rules and leads to a kernel of size $O(k^2)$. The second uses linear programming as a subroutine and leads to a kernel of size at most $2k$.
- Once we have a kernel for a parametrized problem we can solve it either by applying brute force to solve the kernel or by applying a more clever technique, such as tree-search to obtain a solution much faster than with brute force.
- I introduced the notion of exact exponential algorithms and showed how to use tree search for the vertex cover problem. I also showed how to solve TSP instances on n vertices exactly in time $O(n^2 2^n)$. The idea is to use dynamic programming to find an optimal permutation of the vertices.

Lecture in Week 19
This is the last Lecture! I will show how to solve the 3rd set of exam problems.

Exercises in Week 19

- Left over problems from previous weeks,
- How will the adversary for sorting answer against the algorithm 'insertion sort' (which works by building the sorted list one element at a time and then inserting the next element at its right place in the current list) when sorting 8 elements? Hint: denote the input elements by $x_1, x_2, \ldots, x_8$ and consider how the adversary will sift these elements down the bag structure while the algorithm runs.
- A digraph is an **in-tournament** if whenever u, v, w are distinct vertices and $u \to w$ and $v \to w$ are arcs then there must be an arc between u and v (the direction is not important). We say that a graph G can be **oriented** as an in-tournament if we can assign an orientation to each edge of G so that the resulting digraph D is an in-tournament. Define the problem **IN-TOURNAMENT_ORIENTABLE** as follows. The input is a graph G and the output is 1 if G can be oriented as an in-tournament and 0 otherwise. Show how to formulate this problem as an instance of 2-SAT. That is, you must find a way of making a formula for a given graph G so that this formula is satisfiable if and only if G can be oriented as an in-tournament. Also consider the complexity of this transformation. Is it polynomial? Hint: consider a reference orientation D of G. Make a variable x_i for each arc a_i of D and let $x_i = 1$ mean that you keep the orientation whereas $x_i = 0$ should mean that you reverse the orientation. Now construct a set of clauses that will force all violating triples u, v, w to be corrected.

- Show that the following problem is NP-hard: Given a connected graph G. Find a spanning tree T so that the maximum degree of a vertex in T is as small as possible. The degree of a vertex v in T is the number of edges in T which have v as one of their end vertices.

- Recall that a graph $G = (V, E)$ is 2-edge-connected if and only if no matter how we partition V into disjoint sets $U, V \setminus U$ there are at least two edges from E with one end in each of U and $V \setminus U$. Show that the following problem is NP-hard: Given a 2-edge-connected graph $G = (V, E)$. Find a minimum size subset $E' \subseteq E$ so that the graph $G' = (V, E')$ (i.e. using only the edges from E') is 2-edge-connected.

- Consider the following problem called **E1-2AUG**: The input is a 2-edge-connected graph $G = (V, E)$ a spanning tree $T = (V, F)$ of G and a weight function ω on $E' = E \setminus F$. Find a minimum weight subset $A \subseteq E'$ so that adding these edges to T results in a 2-edge-connected spanning subgraph of G. Let us say that an edge $uv \in E'$ **covers** the edge st of T if st is on the unique path from u to v in T.

  1. Show that adding the edges of a subset $A \subseteq E'$ to T will give a 2-edge-connected graph if and only if every edge in T is covered by at least one edge in A.
  2. Formulate **E1-2AUG** as a set covering problem. Hint consider for each $e \in E'$ the set S_e consisting of those edges of T that are covered by e.
  3. Suppose A is the set of edges of a minimum spanning forest (that is, a minimum spanning tree in each connected component) of the graph $H = (V, E')$ obtained by deleting all edges of T from G. Use the fact that G is 2-edge connected to prove that A covers each edge of T at least once.

- The HITTING SET problem is as follows Given a set S, a collection $X_1, \ldots, X_m$ of subsets of S and a natural number k; Does there exist a subset $S' \subseteq S$ such that $|S'| \leq k$ and $S \cap X_i \neq \emptyset$ for $i = 1, 2, \ldots, m$. Prove that the HITTING SET problem is NP-complete. Hint: reduce VERTEX COVER to HITTING SET.