Institut for Matematik og Datalogi
Syddansk Universitet

April 24, 2024
JBJ

DM553-MM850 – Spring 2024– Weekly Note 12

Important: Obligatory Prerequisite test

Remember that the deadline for handing in this is May 3rd at 10.00. **If you do not hand it in you cannot participate in the oral exam!**

The reason for its existence is that previously we have wasted (sometimes lots) of time and money waiting for people whom we have never seen or heard from to show up at the exam. These ghost students will most likely not hand in the prerequisite test and then we don't need to schedule their exam.

Exam info

A PDF with exam questions and other info can be found on the home page.

Stuff covered in Week 17

- Algorithms and matching lower bounds for finding, maximum and minimum element as well as for finding largest and second largest element and for finding the median. This is based on Baase pages 125-133 and my own notes on lower bounds pages 1-8
- I also covered the deterministic linear algorithm for the selection problem based on Cormen Section 9.3.

Key points

- In adversary arguments for comparison based algorithms we must show a strategy which an adversary can follow while answering queries of the kind: which is the largest of x and y. The aim of the adversary is to force the algorithm to make as many comparisons as possible.
- I showed (see my notes) that the adversary can make such answers efficiently and consistently (that is (s)he will never say that $a > c$ if we can already conclude from previous answers that $c > a$) by orienting a subset of the the edges of the complete graph on n vertices (the size of the input) in such a way that the edges that are already oriented always form an acyclic digraph. Then at the end (when the algorithm claims to be done), the adversary can construct a bad input for the algorithm by assign the values $\{1, 2, \ldots, n\}$ to the numbers according to any acyclic ordering of the final oriented graph. This acyclic digraph corresponds to those arcs for which the algorithm has asked for the result of a comparision between the two elements corresponding to the end vertices of that arc.

- In order to know the median m of a set X of $n = 2k + 1$ distinct numbers one needs to have established k elements that are smaller than m and k that are larger. In the digraph model that we use, this means that we must have a directed path from each vertex in some set S of size k to m and a directed path from m to each of the remaining k vertices of X. The adversary uses this to construct, for each median finding algorithm $\mathcal{B}$, an input that depends of $\mathcal{B}$ and for which $\mathcal{B}$ must to use at least k further comparisons before the information above can be obtained.
- The selection problem can be solved in linear time provided we can find a pivot (the element that we partition around) so that we can always eliminate some constant fraction of the remaining numbers before the next round. We saw a clever way of using a two level approach, one of which involves finding the median of a subset of the elements to ensure that we can always eliminate roughly $\frac{3}{10}$ of the remaining elements.

New stuff in Week 18

I am probably going to cancel the friday lecture, so the stuff below will be covered on April 29.

- 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 (FPT). Based on pages 3-7, 12-14 and 17-22 from the book M. Cygan et al, Parametrized Algorithms, Springer. You can find these pages on the homepage of the course. (Video 25).

Exercises in week 18

- JBJ notes on lower bounds exercises 9-11
- Go through the proof of Corollary 7.2 in JBJs notes and convince yourselves that the adversay can indeed perform the strategy efficiently.
- Exercise 3.10 in Baase
- Cormen 9.3-4 page 223. Hint: think of the adversarys strategy (using arrows)
- Suppose you have an algorithm $\mathcal{A}$ for finding the median of a set S of n distinct integers. You may assume that $\mathcal{A}$ returns the median m and two sets $S_<, S_>$ where $S_<$ ($S_>$) consists of those numbers that are smaller (larger) than m.

Question a: Explain how one can build an algorithm $\mathcal{B}$ which when given a set S of n distinct integers as input; first uses $\mathcal{A}$ to find the median m and the sets $S_<, S_>$ and then, using at most $n-1$ extra comparisons also finds the minimum and the maximum element of S.

Question b: Explain how to modify the linear algorithm for the selection problem in Cormen Section 9.3 so that when the median is found we can construct the sets $S_<, S_>$ without using any further comparisons. Hint: use the arrows. Note that in Cormen arrows go from larger elements to smaller, while they go from smaller elements to larger in my notes. Please use my notation in your solution. You should explain briefly how to maintain the arrow structure during the run of SELECT.

Question c: Now consider running any algorithm $\mathcal{A}'$ for the median problem while using arrows to keep track of conclusions from comparisons made by $\mathcal{A}'$, where we use my notation, that is, there is an arrow directly from x to y if $\mathcal{A}'$ compared x to y and found that $x < y$.

Explain how you can use this arrow structure to decide (without making further comparisons) which elements can be the minimum (maximum) element of S. Note that you must exclude those elements which we can conclude cannot be either the max or the min element.

Question d: Explain why you can very often use the arrows produced while running $\mathcal{A}'$ to find both the minimum and maximum elements with much less than $n-1$ extra comparisons after $\mathcal{A}'$ has stopped. Hint: is it possible for some input that there is only one candidate for min (max)?