Soham Chatterjee
Email: soham.chatterjee@tifr.res.in
Course: Algorithms

Assignment - 1
Dept: STCS
Date: September 25, 2024

Problem 1 P2 (10 marks)

Show that $n-1$ comparisons are necessary and sufficient to find the minimum element in an unsorted array of n elements.

Solution:

■

Problem 2 P3 (15 marks)

Show a comparison based algorithm for finding the minimum and maximum in an unsorted array of n elements using $\left\lfloor \frac{3n}{2} \right\rfloor - 2$ comparisons. Also show that $\left\lfloor \frac{3n}{2} \right\rfloor - 2$ comparisons are necessary to find the minimum and maximum.

Solution:

■

Problem 3 P4 (10 marks)

Let $G = (V, E)$ be a directed acyclic graph $G = (V, E)$. Additionally, you are given a nonnegative, integral weight w_e on each edge $e \in E$, and two special vertices $s, t \in V$. Give an algorithm to find a max-weight path from s to t.

Solution:

■

Problem 4 P5 (15 marks)

Given a matroid $(S, \mathcal{I})$, show that $(S, \mathcal{I}')$ is also a matroid, where $A \in \mathcal{I}'$ if $S \setminus A$ contains a maximal independent in $\mathcal{I}$.

Solution:

■

Problem 5 P6 (15 marks)

In class, we showed that if $(S, \mathcal{I})$ is a matroid, then for any nonnegative weights w no the elements of S, the greedy algorithm obtains a maximum weight independent set. Show that this is only true if $(S, \mathcal{I})$ a matroid. That is, for a fixed downward-closed set system $(S, \mathcal{I})$, if the greedy algorithm obtains a maximum weight element of $\mathcal{I}$ for every assignment of nonnegative weights to elements of S, then $(S, \mathcal{I})$ is a matroid.

Solution: ■

Problem 6 P7 (10 marks)

Exercise 10.4-6 (on tree representations with pointers) from CLRS.

Solution: ■

Problem 7 P8 (10 marks)

Given a directed graph $G = (V, E)$ with weights on the edges, and which has a negative-weight directed cycle that is reachable from the source s, Give an efficient algorithm to list the vertices of such a cycle.

Solution: ■

Problem 8 P9 (15 marks)

Let us modify the "cut rule" (in the implementation of decrease-key operation for a Fibonacci heap) to cut a node x from its parent as soon as it loses its 3rd child. Recall that the rule that we studied in class was when a node loses its 2nd child. Can we still upper bound the maximum degree of a node of an $n-$node Fibonacci heap with $O(\log n)$.

Solution: ■

Problem 9 P10 (15 marks)

The following are Fibonacci-heap operations: *extract-min*$(\cdot)$, *decrease-key*$(\cdot,\cdot)$, and also *create-node*(x,k) which creates a node x in th root list with key value k. Show a sequence of these operations that results in a Fibonacci heap consisting of just one tree that is a linear chain of n nodes.

Solution: ■