Extra set of problems

Assigned: March 4 Will not be collected/graded

Variations of the following problems might appear on the midterm. No solutions for this set of problems will be posted (unless a problem becomes part of a later homework).

Problem 1 Show that there is no comparison sort whose running time is linear for at least half of the \( n! \) inputs of length \( n \). What about a fraction of \( 1/n \) of the inputs of length \( n \)? What about a fraction \( 1/(2n) \)?

Problem 2 Our SELECTION algorithms uses partitioning into groups of 5, computing the median in each group, followed by two recursive calls (with some work between them). Analyze the running time for two variants of this algorithm, where the groups are of size 3, and of size 7 respectively.

Problem 3 Given an edge-weighted undirected graph \( G = (V, E, w) \), and a spanning tree \( T \) in \( G \), define the \textit{width} of \( T \) to be the weight of its minimum-weight edge.

Describe an efficient algorithm that, given \( G \), finds a spanning tree \( T \) of maximum width. Present pseudocode, analyze the running time and prove correctness.

Problem 4 We describe below a data structure that maintains the transitive closure of a directed graph while arcs (directed edges) are added to the graph.

Formally, a set of vertices \( V \) is given (with \( |V| = n \)), and arcs \( e_1, e_2, \ldots, e_m \) become available one by one (\( e_i \) is not known before computing \( R_{i-1} \), defined below). Let \( G_i = (V, E_i) \), where \( E_0 = \emptyset \) and \( E_i = E_{i-1} \cup e_i \). Let \( R_i \), a \( n \times n \) matrix, have \( R_i[u, v] = 1 \) if \( u \) has a directed path to \( v \), and \( R_i[u, v] = 0 \) otherwise. Thus \( R_i \) stores the transitive closure of \( G_i \).

Note that \( R_0 \) has entries that are 1 only on the main diagonal.

1. Give a series of instances (one for each \( n \)) such that there exists an \( i \) with the number of entries 1’s in \( R_i \) being \( \Omega(n^2) \) higher than the number of entries 1 in \( R_{i-1} \).

2. Consider however the code

\[
\text{ADD}(e_i) \quad \text{where the tail of } e_i \text{ is } u \text{ and the head of } e_i \text{ is } v:
\]

1. for all \( x \in V 
\]
2. if \( R[x, u] = 1 \) AND \( R[x, v] = 0 
\]
3. for all \( y \in V 
\]
5. \( R[x, y] \leftarrow \max(R[x, y], R[v, y]) 
\]

Prove that if \( R = R_{i-1} \) before the code is executed, then \( R = R_i \) after the code is executed.

3. Use the first part of this problem to show that \( \text{ADD}(e_i) \) may have running time \( \Omega(n^2) \).

4. Prove that despite this, the running time of \( m \) operations \( \text{ADD}(.) \) is \( O(nm + n^3) \).

Problem 5 For Fibonacci Heaps, the rule for cutting during the cascade is to cut a node from its parent after it loses 2 children. What if instead, the cut is done after losing one child? Will the degree of the root still be provable \( O(\log n) \)? Same question, if instead of 2 above one uses an integer \( k > 2 \).