Fundamental Algorithms, Sample Final, Fall 2022

Answer all questions. The exam period lasts 1 hour 50 minutes. 1 hour 45 minutes are for taking the exam; the remaining 5 minutes are for taking a photo of your answers. Then, in the next hour, please upload your solution to Gradescope (linking your answers to questions). Please hand in your written work too. We will conduct spot checks to make sure the written work and the uploaded answers are the same.

1. (**5 pts.**) State whether the following assertions are True (T) or False (F).

a. $100n^4 = O(n^5)$.

b. $2^n + 1000n^2 = \Theta(2^n)$.

c. $3^n = \Theta(4^n)$.

d. Which of the following two functions is larger (or answer "both" if they are equal). Note: this question is not asking about asymptotics but about actual values.

$$2^{\log n} \qquad\qquad n+1$$

e. For the following two functions, which one grows to be the asymptotically larger as n tends to infinity. If they remain within constant factors of each other, answer "both".

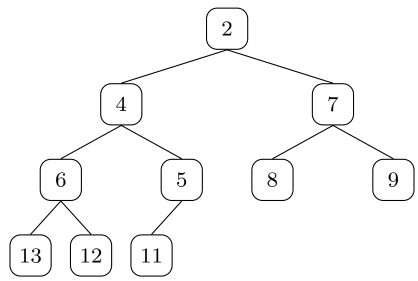
$$(\log n)^2 \qquad\qquad n$$

2. (**5 pts.**) Use the recursion tree method to solve the following recurrence equation. Full credit will be given for a solution that is written as a sum, whether as a closed form or as a sequence of terms (but make sure to show what are the first and last terms and what is the form of the sequence).

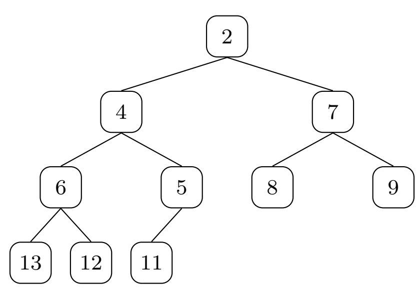
$$\begin{aligned} &S(0) = 0 \\ &S(n) = 2^n + 4S(n-1) \qquad n > 0 \end{aligned}$$

3.(**5 pts.**) Show the effect of the following operations on the given binary heaps.

a. DeleteMin:



b. ReduceKey$(12, 1)$ (meaning to take the item with key 11 and reduce the value of its key to 1).

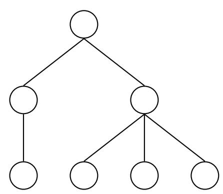


4. (**5 pts.**) What does the statement that the dictionary operations (search, insert, delete) run in expected $O(1)$ time when using a hash table mean?

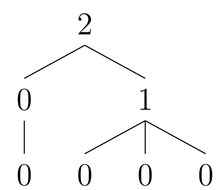
5. (**5 pts.**) Let T be an arbitrary tree so each vertex can have any number of children.

You are to write an algorithm which, for each vertex v, will compute the number of descendants of v including v itself that have two or more children, storing the result in $v.\mathrm{mdsc}$.

The following example tree shows the values to be computed.



Input tree



output tree showing $.\mathrm{mdsc}$ field

Please complete the following procedure. Remember to have a driver procedure or to make an initial call.

procedure Dsc(v,);

for each child w of v **do**

Dsc(w,);

end for

end procedure

Driver procedure/Initial call (**answer here too**):

6.(**5 pts.**) Let $G = (V, E)$ be a dag, and let s and t be two vertices in V. Suppose G is given in adjacency list format. Give a linear time algorithm to determine for every vertex $v \in V$ whether it is on a path from s to t. Justify the running time of your algorithm.

7. (**5 pts.**) Let $G = (V, E)$ be an undirected graph, and let $F \subset E$ be a subset of the edges in E. Let $H = (V, F)$. Suppose H has connected components $C_1, C_2, \ldots, C_k$. Define the connected components graph $K = (U, D)$ of G w.r.t. F as follows. It has vertices $U = \{u_1, u_2, \ldots, u_k\}$. For each i and j, $1 \leq i < j \leq k$, if there is an edge between a vertex in C_i and a vertex in C_j, then there is an edge $(u_i, u_j) \in D$.

Give an algorithm that takes G and F as input and outputs the graph K. Your algorithm should run in linear time. Briefly explain why it is correct and justify your running time.

8. (**10 pts.**) Let $C = \{a_1, a_2, \ldots, a_n\}$ be a collection of n not necessarily distinct positive integers. Let $t = \sum_{i=1}^{n} a_i$.

Your task is to give an algorithm to determine whether C can be partitioned into three disjoint collections C_1, C_2, and C_3, such that the items in C_i sum up to t_i, for $i = 1, 2, 3$. The output of your algorithm is one of True or False.

Your algorithm should run in time $O(t^2 n)$. Less efficient solutions will receive partial credit.

You should express your solution as a recursive formula and then explain how to implement it so as to achieve the desired runtime. Remember to justify the runtime.

9. (**10 pts.**) Let $G = (V, E)$ be a directed weighted graph, with two distinct designated vertices s_a and s_b. Suppose there are racing teams A and B located at the distinct vertices s_a and s_b. They are racing to be the first to reach each of the vertices $v \in V$. However, team A is faster: it travels along an edge (u, v) in time $w(u, v)$, while team B takes time $2 \cdot w(u, v)$. To even things up, team B starts at time 0 and team A starts at time $\bar{t} > 0$. In addition, we guarantee that there will be no ties and that team B will not reach s_b by time $\bar{t}$. There is one additional rule: only the first team to reach a vertex can use the outedges from that vertex.

By modifying Dijkstra's algorithm, determine which team reaches each vertex $v \in V$ first (so the answer for each vertex is one of A and B).

The runtime of your algorithm should be at most a constant multiple of the runtime of Dijkstra's algorithm on graphs with n vertices and m edges. Justify your runtime. Also, briefly explain why your algorithm is correct.

Pseudo-code for the standard Dijkstra's algorithm can be found on the next page.

Hint: It may be helpful to keep an array Rslt$[1 : n]$, where the entry for vertex v has one of the values A, B, N, according to which team has the current shorter path to v, or neither (if the current distance is infinite).

procedure Dijkstra(G, s)
    for each $v \in V$ **do**
        $\text{Dist}[v] \leftarrow \infty$;
    end for
    $\text{Dist}[s] \leftarrow 0$;
    $\text{EnQueue}(Q, V, \text{Dist})$;
    while Q not empty **do**
        $u \leftarrow \text{DeleteMin}(Q)$;
        for each edge (u, v) **do**
            $\text{new\_dist} \leftarrow \text{Dist}[u] + w(u, v)$
            if $\text{new\_dist} < \text{Dist}[v]$ **then**
                $\text{Dist}[v] \leftarrow \text{new\_dist}$; $\text{ReduceKey}(Q, v, \text{Dist}[v])$
            end if
        end for
    end while
end procedure

Other possible topics. Divide and conquer algorithms (e.g., sorting, binary search), search trees, greedy algorithms.