

Practice for Final Exam

- Do not open this quiz booklet until you are directed to do so.
- This final ends at 12:00 P.M.
- This exam is closed book. You may use three sides of handwritten $8\frac{1}{2}'' \times 11''$ crib sheets.
- When the quiz begins, write your name on the top of ***EVERY*** page in this quiz booklet, because the pages will be separated for grading.
- Write your solutions in the space provided. If you need more space, write on the *back* of the sheet containing the problem. Do not put part of the answer to one problem on the back of the sheet for another problem.
- Plan your time wisely. Do not spend too much time on any one problem. Read through all of them first and attack them in the order that allows you to make the most progress.
- Show your work, as partial credit will be given. You will be graded not only on the correctness of your answer, but also on the clarity with which you express it. Be neat.
- When describing an algorithm, describe the main idea in English. Use pseudocode only to the extent that it helps clarify the main ideas.
- Good luck!

Name: _____

Please circle your TA's name and recitation:

Nitin Brian Mihai Yoav

10am 11am 12pm 1pm 2pm

Problem 1. Short questions

In the following questions, fill out the blank boxes. In case more than one answer is correct, you should provide the **best known** correct answer. E.g., for a question “Sorting of n elements in the comparison-based model can be done in time”, the best correct answer is $O(n \log n)$. No partial credit will be given for correct but suboptimal answers. However, you **do not** have to provide any justification.

- (a) Consider a modification to QUICKSORT, such that each time PARTITION is called, the median of the partitioned array is found (using the SELECT algorithm) and used as a pivot.

The worst-case running time of this algorithm is:

- (b) If a data structure supports an operation `foo` such that a sequence of n `foo`'s takes $\Theta(n \log n)$ time to perform in the worst case, then the amortized time of a `foo`

operation is Θ $\left(\begin{array}{|c|} \hline \text{ } \\ \hline \end{array} \right)$, while the actual time of a single `foo` operation could

be as high as Θ $\left(\begin{array}{|c|} \hline \text{ } \\ \hline \end{array} \right)$.

- (c) Does there exist a polynomial time algorithm that finds the value of an $s-t$ minimum cut in a directed graph? (Yes or No)

Problem 2. True or False?

For each of the questions below, circle either **T** (for True) or **F** (for False). **No explanations are needed.** Incorrect answers or unanswered questions are worth zero points.

- T F** By the master theorem, the solution to the recurrence $T(n) = 3T(n/3) + \log n$ is $T(n) = \Theta(n \log n)$.
- T F** Computing the median of n elements takes $\Omega(n \log n)$ time for any algorithm working in the comparison-based model.
- T F** Every binary search tree on n nodes has height $O(\log n)$.
- T F** Given a graph $G = (V, E)$ with cost on edges and a set $S \subseteq V$, let (u, v) be an edge such that (u, v) is the minimum cost edge between any vertex in S and any vertex in $V - S$. Then, the minimum spanning tree of G must include the edge (u, v) . (You may assume the costs on all edges are distinct, if needed.)
- T F** Let T be a minimum spanning tree of G . Then, for any pair of vertices s and t , the shortest path from s to t in G is the path from s to t in T .
- T F** If a problem L_1 is polynomial time reducible to a problem L_2 and L_2 has a polynomial time algorithm, then L_1 has a polynomial time algorithm.

Problem 3. True or False and Justify.

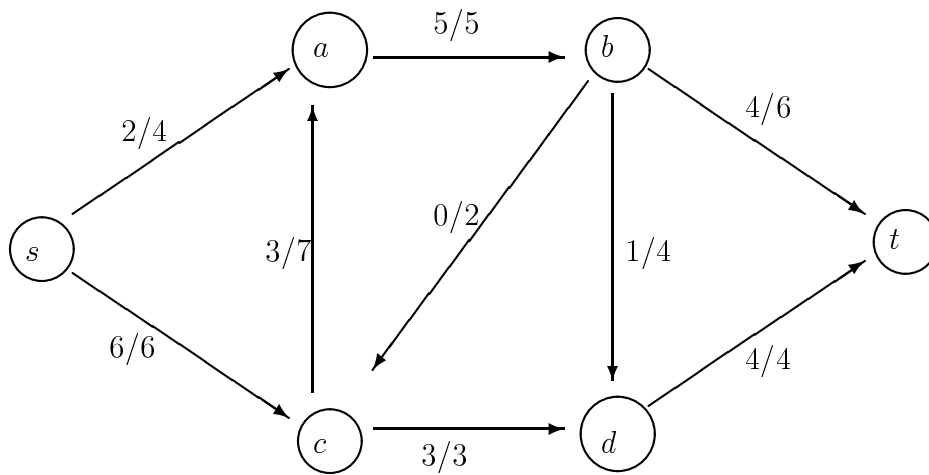
For each of the questions below, circle either **T** (for True) or **F** (for False). Then give a brief **justification** for your answer. Your answers will be evaluated based on the justification, and **not** the **T/F** marking alone.

T F There exists a data structure to maintain a dynamic set with operations $\text{Insert}(x,S)$, $\text{Delete}(x,S)$, and $\text{Member?}(x,S)$ that has an expected running time of $O(1)$ per operation.

T F The total amortized cost of a sequence of n operations (i.e., the sum over all operations, of the amortized cost per operation) gives a lower bound on the total actual cost of the sequence.

T F Let $G = (V, E)$ be a weighted graph and let M be a minimum spanning tree of G . The path in M between any pair of vertices v_1 and v_2 must be a shortest path in G .

T F The figure below describes a flow assignment in a flow network. The notation a/b describes a units of flow in an edge of capacity b . True or False: The following flow is a maximal flow.



Problem 4. Typesetting

Suppose we would like to neatly typeset a text. The input is a sequence of n words of lengths l_1, l_2, \dots, l_n (measured in the number of fixed-size characters they take). Each line can hold at most P characters, the text is left-aligned, and words cannot be split between lines. If a line contains words from i to j (inclusive) then the number of spaces at the end of the line is $s = P - \sum_{k=i}^j l_k - (j - i)$. We would like to typeset the text so as to avoid large white spaces at the end of lines; formally, we would like to minimize the sum over all lines of the square of the number of white spaces at the end of the line. Give an efficient algorithm for this problem. What is its running time?

Problem 5. Princess of Algorithmia (2 parts)

Political upheaval in Algorithmia has forced Princess Michelle X. Goewomans to vacate her royal palace; she plans to relocate to a farm in Nebraska. The princess wants to move all of her possessions from the palace in Algorithmia to the Nebraskan farm using as few trips as possible in her Ferrari. For simplicity, let us assume that the princess's Ferrari has size 1, and that her n possessions x_1, x_2, \dots, x_n have real number sizes between 0 and 1. The problem is to divide the princess's possessions into as few carloads as possible, so that all her possessions will be transported from Algorithmia to Nebraska and so that her Ferrari will never be overpacked. It turns out this problem is NP-hard.

Consider the following **first-fit** approximation algorithm. Put item x_1 in the first carload. Then, for $i = 2, 3, \dots, n$, put x_i in the first carload that has room for it (or start a new carload if necessary). For example, if $x_1 = 0.2$, $x_2 = 0.4$, $x_3 = 0.6$, and $x_4 = 0.3$, the first-fit algorithm would place x_1 and x_2 in the first carload, x_3 in the second carload, and then x_4 in the first carload (where there is still enough room). Note that all decisions are made offline; we divide the princess's n possessions into carloads before any trips have actually been made.

- (a) The princess's charming husband, Craig, has asserted, "The first-fit algorithm will always minimize the total number of car trips needed." Give a counterexample to Craig's claim.

- (b) Prove that the first-fit algorithm has a ratio bound of 2. (*Hint:* How many carloads can be less than half full?)

Problem 6. Division of Power in King Arthur's Court (2 parts)

A large number of problems that we now refer to as optimization problems, actually date back to King Arthur's time. King Arthur's court had many knights. They were the source of many of his problems, some computational and others not. King Arthur often sought the advice of his powerful wizard — Merlin — to solve these computational problems. It is rumored that the mighty Merlin underwent a voyage in a time machine, took 6.046 in the Fall of 2000 and used these skills to solve many of the challenges posed to him.

In the two parts of this question, we'll see some of the problems posed to him.

(a) Ruling the land:

King Arthur's court had n knights. He ruled over m counties. Each knight i had a quota q_i of the number of counties he could oversee. Each county j , in turn, produced a set S_j of the knights that it would be willing to be overseen by. The King sets upon Merlin the task of computing an assignment of counties to the knights so that no knight would exceed his quota, while every county j is overseen by a knight from its set S_j .

- (i) Show how Merlin can employ the Max-Flow algorithm to compute the assignments. Describe the running time of your algorithm. (You may express your running time using function $F(v, e)$, where $F(v, e)$ denotes the running time of the Max-Flow algorithm on a network with v vertices and e edges.)

- (ii) Merlin runs his algorithm (from Part (i)) but the algorithm does not produce an assignment that fits the requirements. King Arthur demands an explanation. Merlin explains his algorithm to King Arthur, and King Arthur is convinced that the algorithm is correct. But he does not believe Merlin has executed this algorithm correctly on the given instance of the problem. He wants Merlin to convince him that no assignment is possible in this particular case. Based on your understanding on Max-Flow, what proof would you suggest? (Note that your proof strategy should work for every instance of the problem for which a satisfactory assignment does not exist.)

(b) **Conflict resolution:** At an annual meeting of the knights, King Arthur notes that several of knights have started to develop conflicts. Sending his undercover agents, he has managed to find out a list of all possible duelling pairs. He now wishes to exile a small set of knights away so that no duelling pair remains at the annual meeting. Merlin is now set with this task of finding out which knights to send away. Being a perfectionist, Merlin wishes to find the smallest set he could send into exile. However, after much thought, he is unable to find the optimal solution using any efficient algorithm.

(i) Explain why? (I.e., what problem is Merlin trying to solve and why is he unable to do it.)

(ii) How could Merlin weaken his goals to find a reasonable solution?