Massachusetts Institute of Technology
6.046J/18.410J: Introduction to Algorithms
Professors Madhu Sudan and Seth Teller

Final Exam
December 19, 2000

Final Exam

- Do not open this final exam until you are directed to do so.
- This exam ends at 4:30 P.M. It contains four problems, each with multiple parts. There are **9** pages to the quiz. You have 180 minutes to earn **180** points.
- This exam is closed book. You may use two handwritten $8\frac{1}{2}'' \times 11''$ crib sheets.
- When the exam begins, write your name on the top of every page of this exam booklet.
- Write your solutions in the space provided. If you need more space, write on the *back* of the sheet containing the 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.
- Good luck!

Problem	Points	Grade
1	40	
2	40	
3	60	
4	40	
Total	180	

Name: ______________________________

Please circle your TA's name and recitation:

Ted **Jon** **Leo** **Alantha**

9am **10am** **11am** **12pm** **1pm** **2pm**

Problem 1. True or False? [40 points] (8 parts)

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

T F The **best case** running time for INSERTION SORT to sort an n element array is $O(n)$.

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.

T F Computing a^b takes exponential time in n, for n-bit integers a and b.

Problem 2. True or False and Justify. [40 points] (4 parts)

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. Each question is worth ten points.

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

Justification:

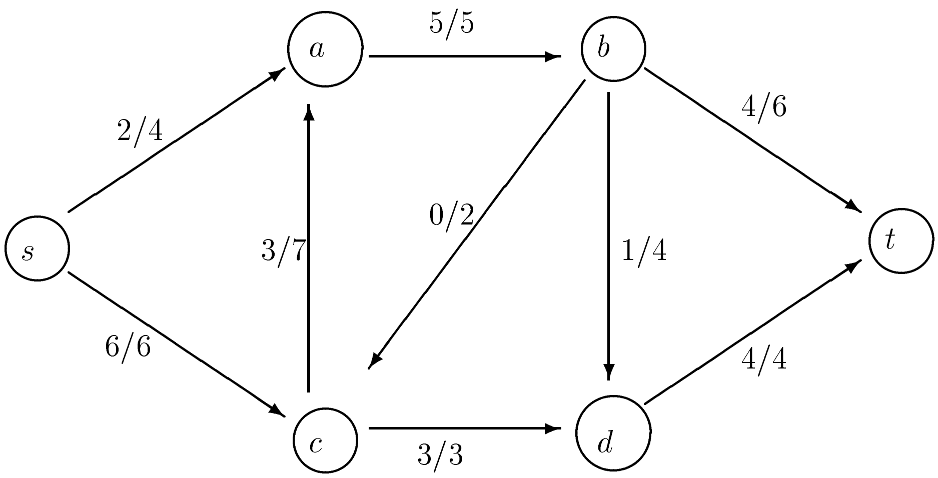
T F The total amortized cost of a sequence of n operations (i.e., the sum over all operations, of the amortized cost per operation) gives a lower bound on the total actual cost of the sequence.

Justification:

T F Any sequence of n increments and decrements on a binary counter takes $O(n)$ time.
Justification:

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.



Justification:

Problem 3. Division of Power in King Arthur's Court [60 points] (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: (40 points)

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: (20 points) 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?

Problem 4. Independent Set in Trees [40 points] (2 parts)

We are given a rooted binary tree T on vertices V with non-negative weights on *vertices*. Let $w(u)$ denote the weight of a vertex u. Our goal is to find the largest **weighted independent set** of vertices in T. I.e., a set $S \subseteq V$ that maximizes $\sum_{u \in S} w(u)$, such that no two vertices in S share an edge in T. In this question you will be asked to solve this problem recursively.

For a vertex u, let

$A(u)$ = Weight of the maximum weighted independent set in the subtree rooted at u. (Note that this independent set may or may not contain u.)

$B(u)$ = Weight of the maximum weighted independent set in the subtree rooted at u, among independent sets that do NOT include u.

(a) For a node u of the binary tree, let $\text{left}(u)$ be the left child of u, and let $\text{right}(u)$ be the right child of u. Given $w(u)$, $A(\text{left}(u))$, $A(\text{right}(u))$, $B(\text{left}(u))$, and $B(\text{right}(u))$, give simple formulae describing $A(u)$ and $B(u)$.

(b) Use the relations of the previous part to give an efficient algorithm to compute $A(\mathrm{root}(T))$. Be sure to analyze the running time of your algorithm.