CS 5633 Analysis of Algorithms – Fall 16

Final Exam

NAME:

- This exam is closed-book and closed-notes, and electronic devices such as calculators or computers are not allowed. You are allowed to use a cheat sheet (half a single-sided letter paper).
- Please try to write legibly – if I cannot read it you may not get credit.
- **Do not waste time** – if you cannot solve a question immediately, skip it and return to it later.

1) Recursion Trees		10
2) Master Theorem		9
3) Search Trees		9
4) Greedy Algorithms		13
5) Dynamic Programming		15
6) Graph Algorithms		10
7) NP-Completeness		17
8) Max Flow		17
		100

1 Recursion Trees (10 points)

Use a recursion tree to generate a guess of what $T(n) = 2T(n/2) + cn^2$ solves to.

2 Master Theorem (9 points)

Solve the following recurrences using the master theorem. Justify your answers shortly (i.e. specify ϵ and check the regularity condition if necessary).

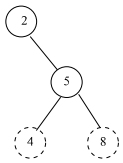
1. $T(n) = T(n/2) + \sqrt{n}$

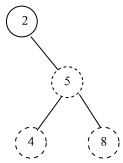
2. $T(n) = 4T(n/2) + n^2$

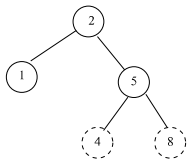
3. $T(n) = 9T(n/3) + n^2 \log n$

3 Search Trees (9 points)

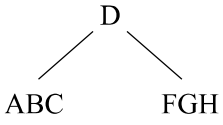
1. Which of the following trees are legal red-black trees, where solid means the node is black and dotted means it is red? If it is not legal, then give a one or two sentence explanation for why it is not a legal B-tree.

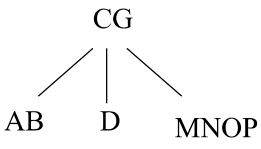


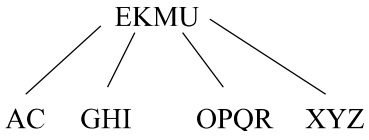




2. Which of the following trees are legal B-trees for $t = 3$? If it is not legal, then give a one or two sentence explanation for why it is not a legal B-tree.







4 Greedy Algorithms (13 Points)

Suppose we are given a collection of n intervals $I_1, I_2, \ldots I_n$ with integer endpoints. Let I_j^ℓ denote the left endpoint of I_j, and let I_j^r denote the right endpoint of I_j. We are also given a set S of integers, such that each interval I_j contains at least one of the integers in S. We say a subset $P \subseteq S$ *stabs* the n intervals if each interval contains some point of P. Give a greedy algorithm which computes a set of points with minimum cardinality that stabs the n input intervals. (Note that this is similar to the problem on the review, but now the set of possible point locations is pre-determined. You cannot choose points to stab anywhere you would like.) Why is your algorithm correct? What is the running time of your algorithm?

- Example: Let $I_1 = [1, 10], I_2 = [4, 15]$, and $I_3 = [8, 20]$, and let $S = \{2, 9, 14\}$. The subset $P_1 = \{2, 14\}$ stabs the intervals (I_1 contains 2 and I_2 and I_3 contain 14). However the set $P_2 = \{9\}$ also stabs the intervals (all three intervals contain 9), and we would prefer P_2 to P_1 since it contains fewer points.

5 Dynamic Programming (15 points)

In this problem, we consider the weighted case of the previous problem. That is, each integer $s_j \in S$ has a weight w_j, and now our goal is to compute a subset of S of *minimum weight* that collectively stabs all n intervals. In this problem we will consider giving a dynamic programming algorithm for this problem.

1. In the previous problem, you were to give a greedy algorithm to minimize the number of points needed to stab all intervals. Show that in this problem, such an algorithm may not return an optimal solution. That is, give an example where a minimum-cardinality solution may not be a minimum-weight solution.

2. Since the greedy algorithm no longer works, consider a brute force algorithm which considers every subset of points and takes the best, feasible solution. What is the running time of this algorithm?.

3. Let $a[i]$ denote the cost of an optimal solution when considering only the first i intervals (you can specify the order of the intervals in any way you want as long as you make it clear what the order is if it is important). Give a recursive definition for $a[i]$. Do not forget the base case.

6 Graph Algorithms (10 Points)

A graph G is *bipartite* if its vertices can be colored red and blue such that every edge connects a red vertex and a blue vertex. Note that a triangle is not a bipartite graph, because every way we color the vertices red and blue, there must be an edge connecting two red vertices or an edge connecting to blue vertices.

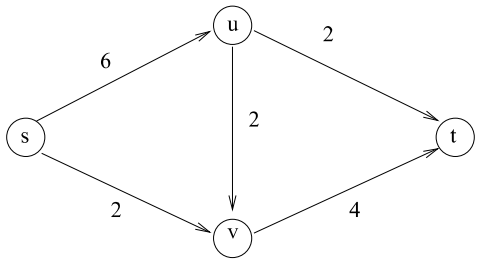
Give an algorithm to determine whether or not an input graph G is bipartite. Your algorithm's running time should be as fast as possible. Briefly argue why your algorithm is correct.

7 NP-Completeness (17 points)

Suppose we are managing a company with n employees, and each employee is in at least one of m mailing lists. We want to "broadcast" an email to each of the empoloyees through one or more of the mailing lists. That is, we want to choose a subset of the mailing lists to send the email to so that each employee recieves the email at least once (it is okay for an employee to receive the email more than once). We would like to know if it is possible to reach all empoloyees using at most k mailing lists.
Show that this problem is NP-Complete. That is, show that it is in NP, and show that it is NP-hard.

8 Max Flow (17 points)

1. Use Ford-Fulkerson to compute a maximum flow in this flow network. What is the value of the maximum flow? What is a minimum cut of the flow network?



2. Suppose that there are n psychology students who are hoping to obtain an internship at one of m intersnship sites. Each student has submitted a list of sites they are willing to be matched with, and each site submits a list of students that they are willing to take on. Each student can match with at most one site, and each site j can take s_j students. A student should be matched to a site only if both (1) the student listed the site and (2) the site listed the student. We would like to compute such an assignment of students to sites so as to maximize the number of matchings. Show that this problem can be reduced to computing a maximum flow in an appropriately defined flow network. You do not need a formal proof as to why it works, just describe the construction of the flow network.