Final Exam

COMP 251 Algorithms and Data Structures
Tues. April 15, 2014 9 AM

Examiner: Michael Langer
Associate Examiner: Joseph Vybihal

LASTNAME: ________________ FIRSTNAME:_____________________ ID: ___________________________

Instructions:

- This is a closed book exam.
- You may use up to five double sided CRIB sheets.
- No electronic devices are allowed.
- If your answer does not fit on a page, then use the reverse side and indicate that you have done so.

.

question	points	score
1	4	
2	4	
3	3	
4	4	
5	4	
6	4	
7	3	
8	4	
9	3	
10	4	
11	4	
12	3	
13	3	
14	3	
TOTAL	50	

1. (4 points)

a) Build an AVL tree by inserting D, E, F, G, A, B, C in that order. Assume the usual comparison ordering on letters, namely A < B < C etc. Assume the tree is initially empty.

b) Same question as (a) but now build a 2-3 tree.

2. (4 points)

a) Consider a list of n integers. Suppose we want to know the integer in that list that occurred the most times, i.e. we allow for repeats. Describe how to use a hash table to solve this problem. Give non-trivial i.e. tight O() bounds on the space and time.

b) Consider the following sequence of (key, value) pairs. A hashcode is given for each key. Assume these entries are **put** into a hash table with m=10 slots.

(“A”, “abbie”) 79
(“B”, “bob”) 53
(“C”, “carla”) 88
(“D”, “doug”) 59
(“E”, “edie”) 71
(“F”, “freda”) 8
(“G”, “greg”) 20

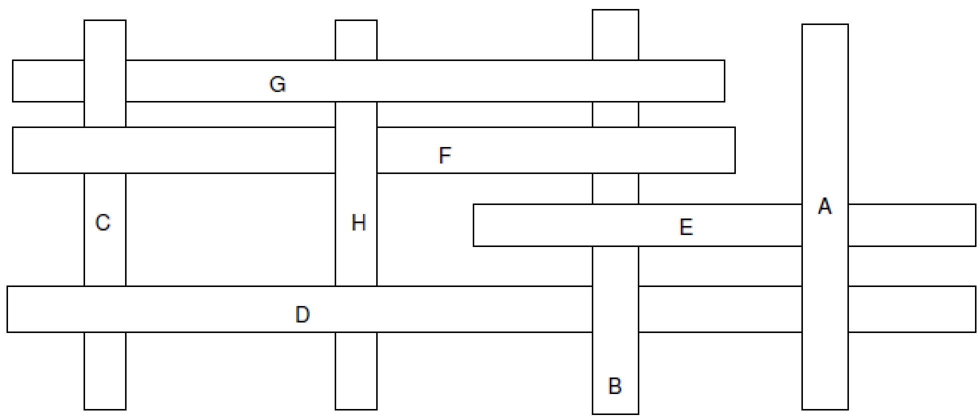
Draw the resulting hash table. Assume closed hashing using *quadratic probing* to handle collisions.

3. (3 points)

The figure below shows a set of rectangles whose overlap relationships can be represented using a directed graph, as follows. Each rectangle is represented in the graph by one vertex, and there is a directed edge in the graph whenever part of one rectangle lies on top of another rectangle. For example, there is an edge from G to H, but not from H to G and there are no edges between A and B since these rectangles do not overlap at all.

a) What are the strongly connected components of this graph ?
Write the answer as partition of the set {A, B, C, D, E, F, G, H }.

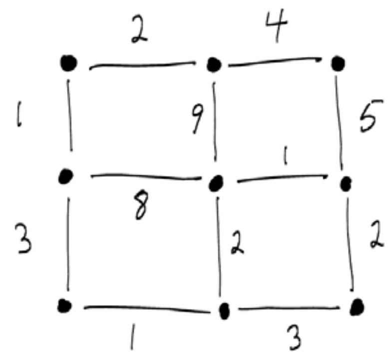
Hint: It will not help you to draw the graph. Instead, solve the problem by visually examining the figure.



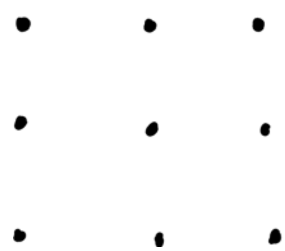
b) Give a topological ordering if one exists. If no topological ordering exists, say why.

4. **(4 points)**

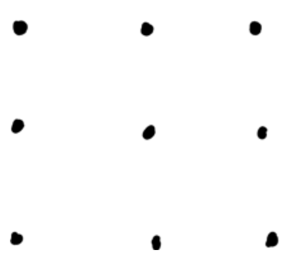
Consider the following graph.



a) Show the edges found by Dijkstra's algorithm, *at the stage* when three vertices (not including the starting vertex) have been reached from the starting vertex. Assume the starting vertex is the center vertex, that is, the vertex with four adjacent vertices. Also, for each vertex in the graph, indicate the length of the shortest known path from s to that vertex *at this stage of the algorithm*.



b) Draw the full minimal spanning tree for the above graph which would be found by Prim's algorithm, starting from the central vertex. Indicate the order in which the edges are added to the tree, namely from 1 to 8.

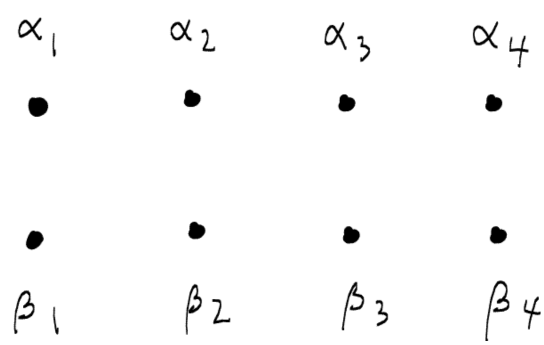


5. **(4 points)**

a) Consider an instance of the stable marriage problem shown below.

A	A's preferences		B	B's preferences
α1	β4 β2 β1 β3		β1	α1 α2 α3 α4
α2	β3 β2 β1 β4		β2	α4 α2 α3 α1
α3	β2 β3 β4 β1		β3	α3 α4 α1 α2
α4	β2 β4 β1 β3		β4	α1 α2 α4 α3

Use the Gale-Shapley algorithm to find a stable matching. Draw the matching below.

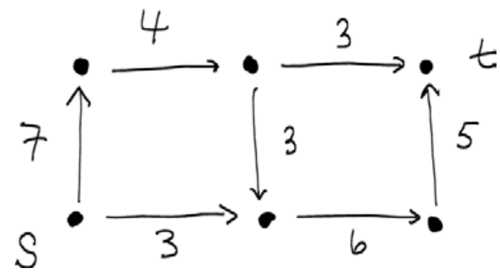


b) Consider an instance of the stable marriage problem in which the Gale-Shapley algorithm yields the same matching when the A's choose as when the B's choose. Can we conclude that there is only one stable matching for this instance? If yes, explain why. If no, give a counterexample.

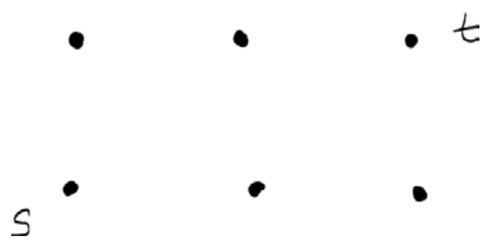
6. **(4 points)**

Consider the following capacity graph for an s-t flow network:

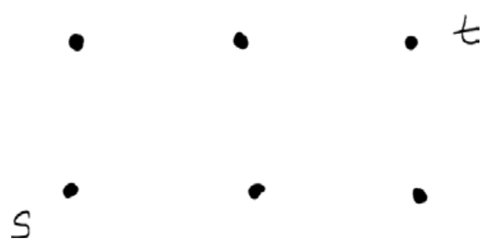
.



a) Draw the graph representing the maximum flow.



b) Draw the residual graph, i.e. after the maximum flow has been computed.



c) Draw the minimum cut. Note: a cut of a graph is a partition of the vertices into two sets.

7. **(3 points)**

Recall the least squares segmentation problem on a set { (xi, yi) : i in 1 to n }. There, the recurrence for the number of segmentations with k segments was f(n,k) = f(n-1,k-1) + f(n-1, k).

The formulation that you have seen allowed for a segment to consist of just one (xi, yi) pair. Consider a new formulation in which we require that each segment has at least two (x,y) pairs. For example, for n=5, the segmentation (*)(****) would not be allowed whereas (**)(***) would be allowed.

Write a recurrence for the number of segmentations with k segments for this new formulation. Briefly explain each term in your recurrence.

Hints:

- Consider all possibilities for what the first segment could contain.
- The base cases are f(1, 1) = 0, f(2,1) = 1, f(3, 1) = 1, .

8. (4 points)

Consider the problem of cutting a rod that is n units long into integer length pieces. Suppose the value (e.g. selling price) for each piece that is j units long is v_j for j in 1 to n. You would like to cut the rod into pieces such that you maximize the total value of all the pieces.

a) Write a recurrence that could be used for solving this problem using dynamic programming.

b) Assume you have found the maximum total value for a problem instance of size n, in particular, you have computed opt(i) for i in 1 to n.
Give pseudocode for a backtracking algorithm that will tell you how many pieces of each size to cut in order to achieve this maximum. Your algorithm only needs to find one solution, i.e. the solution might not be unique.

9. (3 points)

Suppose we place n points in a unit square (1 x 1).

a) Give a non-trivial O( ) bound on the minimum distance δ between any two of the points, as a function of n. Briefly justify your answer.

Note: you are used to thinking of O( ) bounds on run time and memory space, but O( ) can be applied to any quantity that depends on n.

b) Assuming the n points are placed uniformly randomly in the square, give an O( ) bound on the *expected* number of points that are a distance δ (the closest pair distance) or less from any vertical line, for example, the central line used in the initial call to closestPairOfPoints().

10. (4 points)

a) Briefly describe a divide and conquer algorithm for computing the sum of n positive integers. You may assume the integers all have the same number of digits which is a constant.

b) Write out a recurrence for your solution, and identify (circle) which case of the Master method applies.

$$t(n) = a\,t\left(\frac{n}{b}\right) + n^d, \quad t(1) = 1$$

$$t(n) \text{ is } \begin{cases} O(n^d \log_b n), & a = b^d \\ O(n^d), & a < b^d \\ O(n^{\log_b a}), & a > b^d \end{cases}$$

c) Solve the recurrence in (b) using back-substitution. Show your work. Is the divide and conquer approach any faster in the O() sense than the usual way of adding n integers? Justify your answer.

11. (4 points)

a) Give a recurrence for solving the selection problem using a median of median of 7's method. This is analogous to the median of median of 5's, but it partitions the list into groups of 7 elements instead of groups of 5.

b) Show that the solution to the recurrence is O(n). I don't expect a full proof by induction here as given in class. I just expect a manipulation of the terms in the recurrence to get a O(n) bound.

12. (3 points)

Suppose you map n (key,value) entries to a hash table with m buckets. What is the *expected value* of the number of collisions? Use linearity of expectation and justify your answer.

13. (3 points)

Consider an alphabet (sample space) with five elements {a, b, c, d, e} and probabilities:

p(a) = .25, p(b) = .22, p(c) = .32, p(d) = .16, p(e) = .05

a) Construct a Huffman code such that, at each merge step, the child labelled 0 has probability less than or equal to the child labelled 1. For your answer it is sufficient to draw a tree representing the Huffman codes.

b) Decode the bit string 10100100000111.

14. (3 points)

Suppose we have two lists holding n values each, i.e. 2n values in total. Suppose these 2n values are distinct and comparable. Further suppose that we do not have full access to the list but rather we have to request the list elements individually by calling **select**(list, i) for each of the two lists and for any i in 1 to n.

Outline a divide-and-conquer algorithm for finding the median of the 2n values that makes O(log n) calls to select().

<u>Hints:</u>

- It doesn't matter how the select method is implemented here, nor does it even matter how the list is implemented. That said, it might help you to think of each of the lists as being sorted.
- This is a challenging problem and I do not expect correct and complete pseudocode here. Rather I will just look for whether you get the main ideas for how to solve it.