CS 161 – Fall 2015 – Final Exam

Name:

Student ID:

1:

2:

3:

4:

5:

6:

7:

8:

Total:

1. (15 points) Let $H = [24, 21, 18, 15, 12, 9, 6, 3]$ be an array of eight numbers, interpreted as a binary heap with the maximum value at its root (the first element of the array). What would be the contents of H after performing a delete-max operation on this heap?

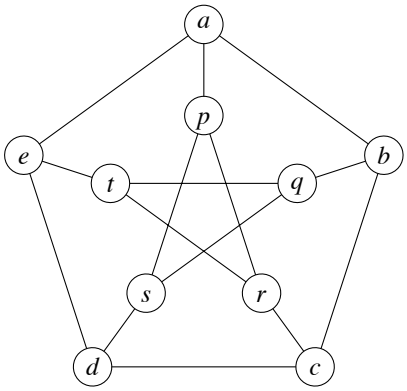
2. (15 points) Suppose we are given n points on a line, and we wish to find one of these points that minimizes the sum of its distances to all the other points. Which of the following algorithms from the class could be used to solve this problem in the most efficient way: heapsort, quickselect, depth first search, the Bellman–Ford algorithm, Graham scan, or the plane-sweep closest-pair algorithm? Explain how the algorithm you choose can be used to solve the minimum-sum-of-distances problem: what input do you give to the algorithm, and how do you produce the best point from its output?

   You may assume that the input is an array of n numbers, which are the x-coordinates of the points, and that their y-coordinates are all zero. Your return value should be the x-coordinate of the optimal point.

3. (15 points) One step of the Karatsuba multiplication algorithm is to split an n-bit input number x into two $n/2$-bit numbers x_1 and x_2, where the binary representation of x_1 consists of the $n/2$ most significant bits of x and the binary representation of x_2 consists of the $n/2$ least significant bits of x. Write assignment statements $x_1 = \ldots$ and $x_2 = \ldots$ that calculate x_1 and x_2 from x, assuming that $n = 32$.

   Your answer should use only addition, subtraction, and bitwise-Boolean operations; do not use the multiplication or division operations. You may use the fact that $2^{16} = 65536$.

4. (15 points) Draw a depth-first search tree for the undirected graph shown below, with the depth-first search starting at vertex b. Each node of your tree should be labeled with the name of one of the graph vertices. When the search loops through the neighbors of any vertex, it should loop through them in alphabetical order.



5. (15 points) Draw an example of a directed graph, with weights on the edges and no negative-weight cycles, containing two vertices s and t such that Dijkstra's algorithm fails to find the correct shortest path distance from s to t. On your graph, what distance does Dijkstra's algorithm find from s to t, and what is the actual shortest path distance from s to t?

6. (15 points) Wythoff's game is played with a single chess rook on an $n \times n$ chessboard. The positions of the board can be numbered by pairs of integers (x, y), where $(0, 0)$ is the square at the bottom left of the board. A move in the game consists of moving the rook either left or down; you lose if the rook is already at $(0, 0)$ and can't move. Define a value $W[i, j]$ to be $+1$ if you can force a win from position (x, y), or -1 if your opponent can force a win no matter how you move from that position. Then these values obey the recurrence

$$W[i,j] = \max \begin{cases} -W[h,j] \text{ for } h < i, \\ -W[i,k] \text{ for } k < j \end{cases}$$

(the maximum is over $i + j$ different earlier values, representing all the places the rook can move) with the base cases $W[0, 0] = -1$, and $W[x, 0] = W[0, x] = +1$ for all $x > 0$. Write pseudocode for a dynamic programming algorithm that takes as input the two numbers i and j and uses this recurrence to compute the value of $W[i, j]$.

Do not use backtracking to find the optimal move; your pseudocode should stop and return $W[i, j]$ once it has been computed.

7. (15 points) Recall that the approximation ratio for a given algorithm, on an optimization problem, is the ratio (quality of the algorithm's output)/(quality of the optimal solution). Consider the 0–1 knapsack problem, in which each item has a weight w_i and a benefit b_i, a solution is a subset of the items with total weight at most equal to the capacity C, and the quality of any solution is the sum of the benefits of the selected items. The greedy algorithm for the fractional knapsack problem chooses items in descending sorted order by b_i/w_i until there is no room for any more items. Suppose that we instead use the same greedy strategy as a heuristic for the 0–1 knapsack problem.

   Give an example of a collection of items with their weights and benefits that causes the approximation ratio of this greedy heuristic to be less than 1/10. Include in your answer the capacity for your example, the solution value found by the greedy heuristic, and the optimal solution value.

8. (15 points) For the set of eight points shown below, what sequence of push and pop operations on its stack would the Graham scan algorithm perform? For each push or pop operation, state which point is being pushed or popped. (Use the version of Graham scan described both in the lecture and in the book, in which the points are sorted in clockwise order around the bottom point.)