

Faculty of Science

**Course:**  CSCI 3070U: Analysis and Design of Algorithms

**Term:** Fall 2017

**Course component:**  Practice Final Exam

# Question 1 – Complexity (34 marks total)

1. (10 marks) Use the recursion tree method to find an estimate for T(n) = 3T(n/4) + lg n, written as a summation of terms representing levels in the recursion tree. Simplify the expression as much as possible in your answer.
2. (10 marks) Find a recurrence, T(n), for the running time of the following pseudo-code, and solve it using any method, showing the result in theta notation (as well as the original T(n)). Show your work, for whichever method you use.

*Note*: Assume that the concatenation operator (+) is θ(n), where n is the size of the left list, and the .. operator is θ(n) where n is the size of the resulting list.

***GRAB-VALS(list)***

1. if list.length == 1 then

2. return list

3. mid = list.length / 2

4. firstHalf = list[0..mid]

5. secondHalf = list[mid+1..list.length-1]

6. firstGrab = GRAB-VALS(firstHalf)

7. secondGrab = GRAB-VALS(secondHalf)

8. grab = firstGrab + secondGrab # concatenate

9. for i = 1 to firstGrab.length do

12. grab = grab + [secondGrab[i] \* firstGrab[i])

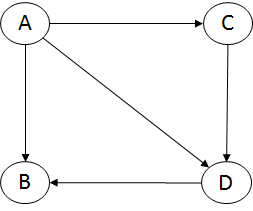
14. end for

15. return grab

1. (10 marks) Using substitution, prove that .
2. (4 marks) Order the following expressions representing running times in terms of their asymptotic order by writing the numbers 1, 2, 3, 4, 5, 6, 7, and 8 in the box beside them (where 1 represents the fastest, and 8 represents the slowest).
   * 18 + 7n – 4n2 + 8n3
   * n log n
   * 2log22n
   * n2 log n
   * n log (n / log n)
   * 3n!
   * 4 log nn
   * n log n2

# Question 2 – Basic Graph Theory (10 marks total)

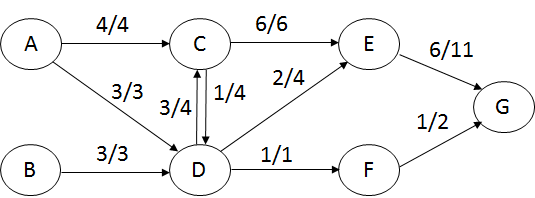
1. (6 marks) Does the following diagram represent a DAG? Explain in one sentence why or why not.



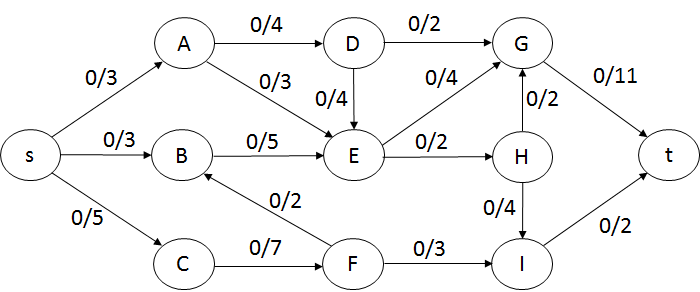
1. (4 marks) Assuming G is a dense graph, what is the running time complexity of BFS(G) with respect to only |V|? (only the answer, in theta notation, is required)

# Question 3 – Max Flow (25 marks total)

1. (10 marks) Find and list the problems with the following flow network, and re-draw the flow network after these problems have been repaired according to the instructor-recommended method.



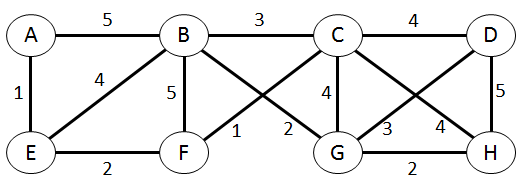
1. (15 marks) Find the max flow of the following network, showing an ordered list of augmenting paths, and each path’s critical edges and bottleneck capacity as you figure out the answer.



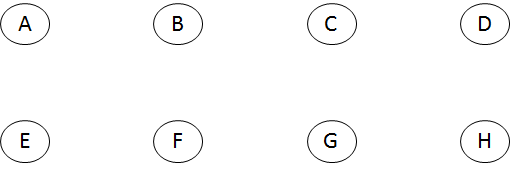
# Question 4 – Minimum Spanning Tree (15 marks)

Find the minimum spanning tree of the following graph, and draw it as a separate diagram. Show the list of edges in the order they were added (as a list of edges) for both Prim’s algorithm (using vertex B as the starting point) and Kruskal’s algorithm (thus, two lists of edges will be included), for comparison.

*Note:* Use the diagram of vertices included below, by augmenting the diagram with the edges that you have added.

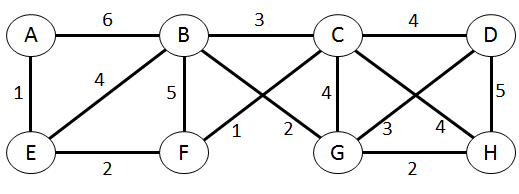


|  |  |
| --- | --- |
| **Algorithm** | **Edges** |
| Prim’s |  |
| Kruskal’s |  |



# Question 5 – Shortest Path (15 marks)

Using Dijkstra’s algorithm, find the shortest path from **B** to all other vertices in the following graph. Show the results in a table with three columns: the vertex, the distance to each vertex, and a path to that vertex.



# Question 6 – Algorithm Strategies (45 marks total)

1. (5 marks) For each of the following algorithms, identify the algorithm strategy used (brute force, divide and conquer, greedy, dynamic programming, or none if none of these fits):
   1. Prim’s Algorithm
   2. Dijkstra’s Algorithm
   3. HeapSort
   4. Kruskal’s Algorithm
   5. MergeSort
2. (20 marks) Design an algorithm to solve the following problem, and write that algorithm in Python, C++, or Java.

*Write a function that takes a list of integers, each of which can be positive, zero, or negative, (called findMaxEvenSumSubset) that finds the subset of those integers such that the sum of those numbers is both an even number and maximal.*

1. (20 marks) In this question, you are to design a dynamic programming algorithm, FIND-LONGEST-PATH, to find the longest path (not shortest path) from any vertex to any other vertex in a directed graph. The length of a path is strictly determined by the number of edges, as edges will not have any weights. Write out your algorithm in C++, Java, or Python.

***Note****: You can assume that the directed graph does not contain any cycles for this question.*

# Question 7 – Theory of Computation (21 marks total)

1. (6 marks) Which algorithm category (P, NP, NP-complete, NP-hard, intractable, non-computable) best describes the following problems?

|  |  |
| --- | --- |
| **Problem** | **Category** |
| Finding a path through an arbitrary graph such that every vertex is visited exactly once. |  |
| To find a forgotten password of a known length, k, for which you know the hashed version of that password involves guessing every possible password, and checking the hash of that guess against the password’s hash. |  |
| Given an arbitrary program, determine the average running time of the program over 100 executions of randomized input(s). |  |

1. (15 marks) Find values that satisfy the following expression, using the method described in the lectures: (¬X ˅ ¬Z) ˄ (Y ˅ Z) ˄ (X ˅ Y) ˄ (¬X ˅ ¬Y)

*Note*: Write all possible values for each variable, if there is more than one.

|  |  |
| --- | --- |
| **Variable** | **Value(s)** |
| X |  |
| Y |  |
| Z |  |

# Question 8 – Minimum Edit Distance (15 marks)

What is the Minimum Edit Distance between the string ‘abc’ and the string ‘ybcab’? Use Levenshtein’s method, and show the complete table that results.

*Note:* You are not required to draw the arrows, as was shown during the lectures. If you draw then anyway, please ensure that they do not obscure the values, which are what will be marked.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | y | b | c | a | b |
|  |  |  |  |  |  |  |
| a |  |  |  |  |  |  |
| b |  |  |  |  |  |  |
| c |  |  |  |  |  |  |