Index Number:							

UNIVERSITY OF MORATUWA

FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

BSc Engineering Honours Degree
2014 Intake Semester 2 Examination

CS2022: DATA STRUCTURES & ALGORITHMS

Time allowed: 2 Hours March 2015

ADDITIONAL MATERIAL: *None*

INSTRUCTIONS TO CANDIDATES:

1. This paper consists of **two (2) Sections** in **12** pages.
2. Answer **all** questions in the question paper itself **in the space provided**. If you make a mistake or need additional space you may attach additional sheets.
3. Section **A** contains **twenty-five (25) MCQ and/or short answer questions**. **Answer these questions in the space provided in the paper**.
4. **For each MCQ question** in section A, there is **only one correct answer** and you are expected to **clearly mark only one choice**.
5. Each question in section A carries 2 marks.
6. Section B contains **two (2)** essay type questions.
7. The maximum attainable mark for each question in section B is given in brackets.
8. The maximum attainable mark for this paper is 100. This examination accounts for 60% of the module assessment.
9. This is a closed book examination.

   NB: It is an offence to be in possession of unauthorized material during the examination.

10. Only calculators approved and labelled by the Faculty of Engineering are permitted.
11. Assume reasonable values for any data not given in or with the examination paper. Clearly state such assumptions made on the script.
12. In case of any doubt as to the interpretation of the wording of a question, make suitable assumptions and clearly state them on the script.
13. This paper should be answered only in English.

SECTION A

1. What is the ADT that is most suitable for implementing a priority queue?
   a) Stack
   b) Queue
   c) Heap
   d) Binary Tree

2. In a depth first forest if node ***v*** is a proper descendent of node ***u***, the relationship between the discovery time (**d**) and finish time (**f**) of the nodes ***u*** and ***v*** is:
   a) **d**[***u***]<**d**[***v***]<**f**[***u***]<**f**[***v***]
   b) **d**[***u***]<**f**[***u***]<**d**[***v***]<**f**[***v***]
   c) **d**[***v***]<**d**[***u***]<**f**[***u***]<**f**[***v***]
   d) There is no relationship between **d**[***u***], **d**[***v***], **f**[***u***] and **f**[***v***]

3. Which of the following statements is true regarding the Single Source Shortest Path (SSSP)?
   a) SSSP of a graph can be found on any graph with or without cycles.
   b) SSSP of a graph can be found on a graph with no negative weight cycles.
   c) SSSP of a graph can be found on a graph with no cycles.
   d) All of the above are true.

4. Which of the following statements is true regarding the hash tables and direct access tables?
   a) Direct access table is more memory efficient than a hash table.
   b) Insertion time on a direct access table is always less than a hash table.
   c) Searching is easy on a hash table, than on a direct access table.
   d) None of the above.

5. Which of the following statements is true regarding the Build Heap and BST?
   a) Build heap algorithm always create a nearly balanced tree.
   b) Build heap algorithm generates a balanced tree only for some input sequences.
   c) Binary Search Tree is always a balanced tree.
   d) None of the above.

6. Which of the following statements is true regarding the Minimum Spanning Tree (MST)?
   a) For each graph there is a unique MST
   b) There can be multiple MSTs to a graph, but their path cost should be similar.
   c) There can be multiple MSTs to a graph, and their path cost can be different.
   d) MST need not to include all the vertices of the graph.

7. What is the advantage of using open addressing over chaining, when implementing a Hash Table?

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

Continued …

8. Minimizing data clustering is a major challenge in Hash Table implementations. Explain what is primary clustering?

....

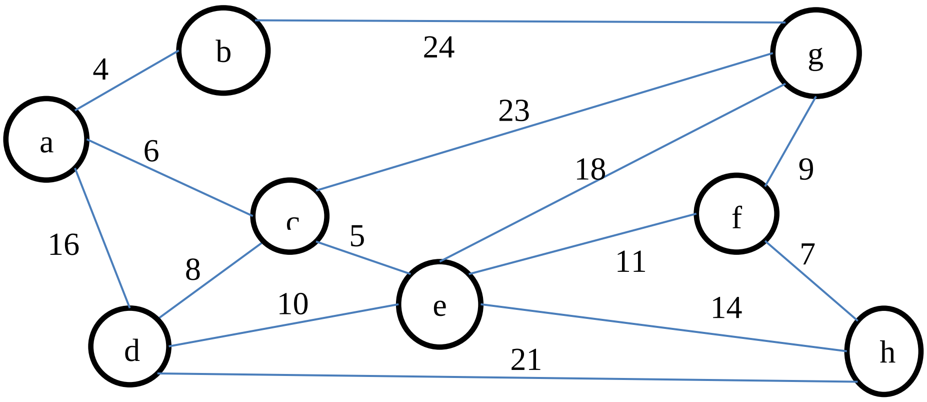
....

....

9. Suggest a solution to avoid primary clustering in a hash table which uses open addressing.

....

10. Draw the MST of the following graph.



Continued …

11. What is the difference between Steiner Minimum Tree and a Minimum Spanning Tree?

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

12. Why does heap sort outperform merge sort in a memory critical applications?

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

13. Which of the following statements is **not** correct?

a) If $f(x) \in \Theta(g(x))$ and $g(x) \in \Theta(h(x))$, then $f(x) \in \Theta(h(x))$ is always true.
b) If $f(x) \in \mathrm{O}(g(x))$ and $g(x) \in \Omega\big(h(x)\big)$, then $f(x) \in \Theta(h(x))$ is always true.
c) If $f(x) \in \omega(g(x))$ and $g(x) \in \omega(h(x))$, then $f(x) \in \Omega(h(x))$ is always true.
d) All three of the above are correct.

14. If $f(x) = 3x^3 + 9x + 27$ and $g(x) = 2x^2 + 4$ and $h(x) = 5x^2 + 5$ which of the following statements is **correct**?

a) $f(x) \in \Omega\big(g(x)\big)$ and $g(x) \in \Theta\big(h(x)\big)$ and $h(x) \in \Theta\big(f(x)\big)$
b) $f(x) \in \omega(g(x))$ and $g(x) \in \Omega(h(x))$ and $h(x) \in O(f(x))$
c) $f(x) \in \Theta\big(g(x)\big)$ and $g(x) \in \omega\big(h(x)\big)$ and $h(x) \in O\big(f(x)\big)$
d) $f(x) \in \Omega(g(x))$ and $g(x) \in \mathrm{o}(h(x))$ and $h(x) \in \omega(f(x))$

15. Which of the following statements is **true** regarding the time complexity of sorting algorithms?

a) Worst case time complexity of Merge sort is same as that of Heap sort
b) Worst case time complexity of Bubble sort is same as that of Quick sort
c) Worst case time complexity of Insertion sort is same as that of Bubble sort
d) All three of the above statements are true

16. Suppose a divide and conquer algorithm solves a problem by dividing the problem into ***m*** sub-problems and then solving ***k*** sub-problems out of it. The recursive function that represents the time complexity of the algorithm, $T(n)$ could be;

a) $T(n) = kT\left(\frac{n}{2}\right) + f(n)$
b) $T(n) = T\left(\frac{n}{k}\right) + f(n)$
c) $T(n) = mT\left(\frac{n}{k}\right) + f(n)$
d) $T(n) = kT\left(\frac{n}{m}\right) + f(n)$

Continued …

17. Which of the following factors affect the running time of a program?

I. Size of the input dataset
II. Nature of the input dataset
III. Memory available in the computer and the memory requirement of the algorithm.

a) I Only.
b) I and II Only
c) II and III Only
d) All three

18. Which of the following algorithm uses dynamic programming approach?

a) Prim's Algorithm
b) Kruskal's Algorithm
c) Dijkstra's Algorithm
d) None of the above

19. Following array is to be sorted using insertion sort.

22	11	44	55	33	66	88	77

The array after first and second iteration of insertion sort are shown below.

After First Iteration:

11	22	44	55	33	66	88	77

After Second Iteration:

11	22	44	55	33	66	88	77

What will be the array after the **fifth** Iteration of the Insertion sort?

11	22	33	44	55	66	88	77

20. **List one (1) advantage** and **one (1) disadvantage** of Insertion sort compared with Quick sort.

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

Answer: Adv- Simple algorithm. Dis - higher time complexity.

21. Explain the reason for the statement "$f(x) \in \omega(g(x))$ implies $f(x) \notin \theta(g(x))$".

……………………………………………………………………………………………

……………………………………………………………………………………………

Continued …

……………………………………………………………………………………………………

Answer: $f(x) \in \omega(g(x))$ indicates that g(x) cannot be a tight lower bound for f(x).

22. In analyzing algorithms, we generally do not consider the best case performance of an algorithm. Explain the reason(s) for not considering the best case analysis.

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

Answer: Best case can be misleading or bogus. It is possible to come up with an algorithm which will have a good best case running time which does not perform well on average.

23. The **in-order** traversal of a balanced binary tree resulted in 1, 2, 3, 4, 5, 6, 7. What will be the **post-order** traversal result of the same tree?

……………………………………………………………………………………………………..

Answer: 4, 5, 2, 6, 3, 7, 1

24. You are given following sorted array and asked to do a binary search for number 22. Write the sequence of numbers that you would compare before finding the number 22 is in the 4th position of the array.

1	8	15	22	25	33	44	62	88	90	99

……………………………………………………………………………………………………..

Answer: 33, 15, (25 ok to have), (ok to have 22 at the end)

25. Draw a flowchart to find the **minimum** number in a given array of numbers.

Continued …

SECTION B

Q1. **[25 *marks*]**

(a) Pseudocode for breath first search is shown below.

```
BFS(G,s)
for each vertex u in V[G] -  {s}
    do color[u] ← white
        d[u] ← ∝
        π[u] ← nil
color[s] ← gray
d[s] ← 0
 π[s] ← nil
Q ← Φ
enqueue(Q,s)
while Q ≠ Φ
    do u ← dequeue(Q)
       for each v in Adj[u]
          do if color[v] = white
             then color[v] ← gray
                d[v] ← d[u] + 1
                π[v] ← u
                enqueue(Q,v)
       color[u] ← black
```

i) What is the total running time of the algorithm? [2 *marks*]

ii) Analyze the BFS algorithm briefly and explain how the above mentioned upper bound of running time is derived. [5 *marks*]

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

(b) Briefly explain why Bellman-Ford algorithm cannot handle negative weight cycled graphs as an input? [3 *marks*]

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

(c) What is the purpose of priority queue in the Dijkstra's Algorithm? [5 *marks*]

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

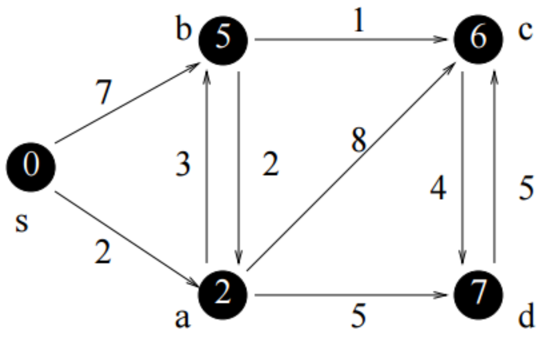
…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

Continued …

(d) The Dijkstra's algorithm runs a while loop for the |V| number of iterations. In the given tables below, fill the Q, adjacency vector (π) and the distance vector (d) after each iteration of the while loop in Dijkstra's algorithm for the following graph starting from S. [10 *marks*]



v	s	a	b	c	d
Initialize : Q = {s,a,b,c,d}					
d[v]	0	∞	∞	∞	∞
Π[v]	nil	nil	nil	nil	nil
Iteration 1 : Q =					
d[v]	0				
Π[v]	nil				
Iteration 2 : Q =					
d[v]	0				
Π[v]	nil				
Iteration 3 : Q =					
d[v]	0				
Π[v]	nil				
Iteration 4 : Q =					
d[v]	0				
Π[v]	nil				
Iteration 5 : Q = Ø					
d[v]	0				
Π[v]	nil				

Continued …

Q2. **[25 *marks*]**

(a)

i. Briefly explain the **two (2) key properties** an optimization problem should have if the dynamic programming approach is applicable to the problem. [4 *marks*]

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

ii. Using these properties explain why dynamic programming solutions would run faster compared to divide and conquer or recursive solutions. [3 *marks*]

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

iii. Explain why greedy solutions would generally run faster compared to dynamic programming solutions. [3 *marks*]

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

Continued …

(b) Give the asymptotic growth in "big oh" notation for the following functions. Please show how you obtained your answer.

i) $T(n) = 1222\, n + (3\times10\text{-}22)\, n^2 + 0.9n \log n$ [2 *marks*]

ii) $T(n) = 3T\left(\frac{n}{2}\right) + \ n \log n.$ [3 *marks*]

(c) Using Merge sort as an example, explain the three main parts of a divide and conquer algorithm. [5 *marks*]

Continued …

(d) Show how the following array will be sorted using the Merge sort. [5 *marks*]

[24, 56, 64, 78, 34, 22, 45, 31, 96]

--- ***End of the Paper*** ---

Continued …