ELL781: Software Fundamentals for Computer Technology

Major Test, Form: $\boxed{\text{B}}$ **(please write this Form ID on the cover page of your answer script)**

Maximum marks: 30

Section 1. Multiple choice questions

Each question may have any number of correct choices, including zero. List (in clear handwriting) the letters corresponding to all choices you believe to be correct (1 mark for each correct choice, -1/2 for each incorrect choice). No justification is required.

1. Consider a graph with 10 vertices and 15 edges, such that no vertex has degree greater than 3. What is the smallest possible number of vertices in a minimum vertex cover of this graph?
   (a) 3
   (b) 4
   (c) 5
   (d) 6
   (e) 9

2. Which of the following recurrences correspond to polynomial-time complexity?
   (a) $T(n) = 2T(n-2) + 2;\ T(1) = T(2) = 1$
   (b) $T(n) = 8T(3n/4) + n^3;\ T(1) = 1$
   (c) $T(n) = T(n-1) + T(n-2);\ T(1) = T(2) = 1$
   (d) $T(n) = nT(n/2);\ T(1) = 1$
   (e) $T(n) = 3T(n/2) + n(n-1);\ T(1) = 1$

3. Consider the problem of determining whether or not there exists a path of length $\leq k$ between *any* pair of vertices in a directed graph. Which complexity class(es) does this problem belong to (assuming $P \neq NP$)?
   (a) NP
   (b) NP-complete
   (c) NP-hard
   (d) P

4. You are given a large network (graph) consisting of data from Facebook: a million vertices corresponding to users, and undirected edges corresponding to friendships between users. Each edge is weighted in inverse proportion to the frequency of interaction between the two friends. You need to find a Minimum Spanning Tree of this network. Which algorithm will be better?
   (a) Kruskal's algorithm should be faster.
   (b) Both algorithms should take about the same time.
   (c) Prim's algorithm should be faster.

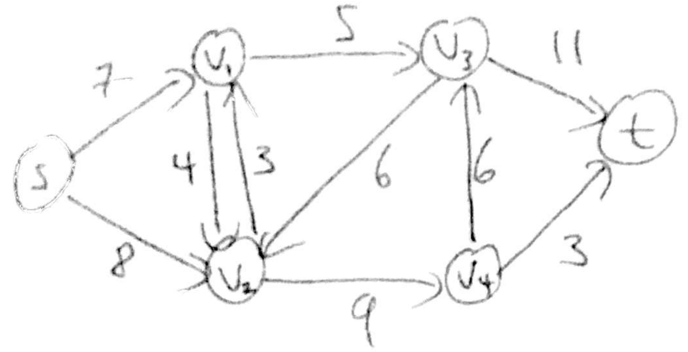
5. Recall the temporal operators $\Box$ (true at all future moments), $\Diamond$ (true at some future moment), and $\circ$ (true at the next moment). Also, recall that $\neg$ is the standard `NOT` operator. Consider the statement $\Box\neg\textit{cash\_shortage}$. Which of the following are *negations* of this statement?
   (a) $\Diamond\textit{cash\_shortage}$
   (b) $\neg\Diamond\neg\textit{cash\_shortage}$
   (c) $\Box\circ\textit{cash\_shortage}$
   (d) $\circ\textit{cash\_shortage}$

Scanned by CamScanner

Section 2. Short Answer Questions

NB: Please read all questions carefully. There may be subtle differences between what the question is asking for and the context in which things have been discussed during the lectures, and these have to be taken into account whilst answering. Working/derivation for all your answers should be shown fully and clearly.

6. Consider the below flow network.



(a) Show the execution of the Edmonds-Karp algorithm (i.e., Ford-Fulkerson with BFS) on the above network, clearly depicting all the steps. **[5]**

(b) In the maximum flow obtained above, what is the flow across the cut $(\{s, v_2, v_4\}, \{v_1, v_3, t\})$? What is the capacity of this cut? **[1]**

(c) Is the cut just examined a minimum cut of the given flow network? If not, find a minimum cut. **[1]**

7. Consider the following randomised approach to search for a value x in an unsorted array A of length n: pick a random integer $i \in 1, 2, ..., n$. If $A[i] = x$, then terminate; otherwise, continue the search by picking a new random index into A, and checking that value. This goes on until either x is found in A, or we have checked every element of A. Note that the index is randomly chosen from the entire range $1, 2, \dots, n$ each time, so a given element may be examined more than once.

(a) Write pseudocode for a procedure that implements the above search strategy. Remember, it should terminate when all the indices into A have been picked. **[3]**

(b) Suppose the value x occurs exactly once in A. What is the expected time complexity of your procedure, as a function of n? Show the derivation clearly. **[2]**

(c) More generally, suppose there are $k > 1$ occurrences of x in A. What is the expected time complexity in this case, as a function of n and k? **[2]**

(d) Suppose x does not occur in A. What is the expected time complexity then? **[2]**

8. In class, we obtained a randomised 8/7-approximation algorithm for the MAX-3-CNF satisfiability problem under the assumption that no clause contains both a variable and its negation. Now re-do the proof without this assumption, showing that the approximation ratio for the given algorithm remains unchanged! **[7]**

Scanned by CamScanner