IMPERIAL COLLEGE LONDON

TIMED REMOTE ASSESSMENTS 2021-2022

BEng Honours Degree in Computing Part I
MEng Honours Degrees in Computing Part I
BEng Honours Degree in Mathematics and Computer Science Part I
MEng Honours Degree in Mathematics and Computer Science Part I
for Internal Students of the Imperial College of Science, Technology and Medicine

This paper is also taken for the relevant assessments for the Associateship of the City and Guilds of London Institute

PAPER COMP40008

GRAPHS AND ALGORITHMS

Friday 13 May 2022, 10:00
Writing time: 80 minutes
Upload time: 25 minutes

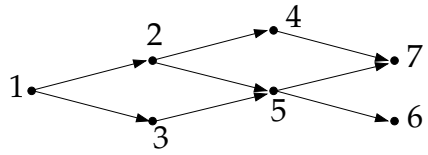
Answer ALL TWO questions
Open book assessment

This time-limited remote assessment has been designed to be open book. You may use resources which have been identified by the examiner to complete the assessment and are included in the instructions for the examination. You must not use any additional resources when completing this assessment.

The use of the work of another student, past or present, constitutes plagiarism. Giving your work to another student to use constitutes an offence. Collusion is a form of plagiarism and will be treated in a similar manner. This is an individual assessment and thus should be completed solely by you. The College will investigate all instances where an examination or assessment offence is reported or suspected, using plagiarism software, vivas and other tools, and apply appropriate penalties to students. In all examinations we will analyse exam performance against previous performance and against data from previous years and use an evidence-based approach to maintain a fair and robust examination. As with all exams, the best strategy is to read the question carefully and answer as fully as possible, taking account of the time and number of marks available.

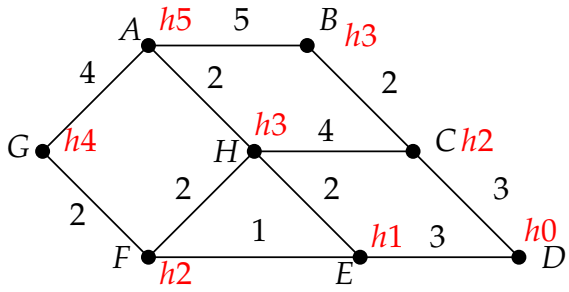
Paper contains 2 questions

1 a Use depth-first search to perform a topological sort on the following directed acyclic graph (DAG). Assume that the adjacency lists are given in ascending numerical order.



State the ordering of the nodes in the topological sort, as well as the order in which nodes are entered and the order in which they are exited. Also draw the DFS tree.

b Use the A* algorithm to find the shortest path from A to D in the weighted graph below, where the values of the heuristic function h are shown in red as $h0$, etc. Draw the corresponding tree, and also state the order in which nodes are added to the tree.



c i) Here is a graph algorithm which iterates through the nodes of an undirected graph and their adjacency lists.

```
count = 0
for x in Nodes:
    count = count+4
    for y in adj[x]:
        count = count+5
return count
```

If this algorithm is applied to a simple graph with n nodes and m arcs, what is the value (of `count`) returned? Give both an exact answer and one using big-O notation.

ii) Here is another graph algorithm which iterates through the nodes of an undirected graph and their adjacency lists, using the recursive procedure `iterate`.

```
procedure iterate(x):
    visited[x] = true
    count = count+3
    for y in adj[x]:
        count = count+2
        if not visited[y]:
            iterate(y)
    count = count+2
# Main program:
count = 0
for x in Nodes:
    visited[x] = false
for x in Nodes:
    count = count+3
    if not visited[x]:
        iterate(x)
return count
```

If this algorithm is applied to a simple graph with n nodes and m arcs, what is the value (of `count`) returned? Give both an exact answer and one using big-O notation. Explain your answer briefly.

d It is desired to buy lengths of wood which will add up to a total length of n units. The cost of a piece of wood of length $k \geq 1$ is $c[k]$, where $c[k] = 0$ means that no piece of length k is available to buy. Write a program (in pseudocode or Java) to compute the minimum cost $M[n]$ achievable for buying lengths of wood which add up to a total length of n units. Your program should return 'not found' if no solution is possible. Here n and k are natural numbers. It is allowed to buy as many pieces of wood of the same length k as desired, provided $c[k] > 0$.

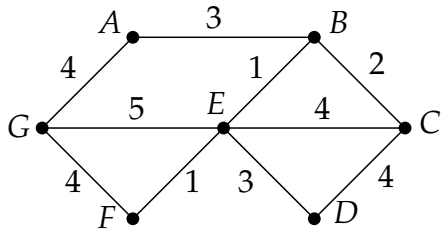
Your program should use a bottom-up (non-recursive) dynamic programming style.

If you wish, you can use the following outline and fill in the blanks ($\cdots$):

```
M[0] = ···
found[0] = ···
for i = 1 to n:
    M[i] = ···
    found[i] = ···
    for j = 1 to i:
        ···
...
```

The four parts carry, respectively, 20%, 20%, 30%, and 30% of the marks.

2a i) Use Prim's algorithm starting from node A to find a minimum spanning tree (MST) for the following weighted graph. Give the MST as a diagram and also state the order in which the nodes are added.



ii) How many distinct MSTs does the graph from (i) have? Explain your answer.

b For each of the following recurrence relations, use the Master Theorem to obtain a solution up to Θ. In each case state the critical exponent E and explain your answer briefly.

i) $T_1(n) = 4T_1(n/2) + 8n$

ii) $T_2(n) = 8T_2(n/4) + 2n \log n$

c Let $n \geq 2$ be an even number, and let L be a list consisting of n distinct numbers, which is ordered as follows. The $n/2$ largest numbers are listed in ascending order, followed by the $n/2$ smallest numbers, again in ascending order. For instance for $n = 8$ the list L could be $5, 6, 7, 8, 1, 2, 3, 4$.

Calculate how many comparisons Insertion Sort makes to sort L.

d Given an undirected graph G, a set $I \subseteq nodes(G)$ is said to be *independent* if no two nodes of I are adjacent.

i) The problem INDN is defined as follows: given an undirected graph G, a node x of G and $k \geq 1$, is there an independent set in G of size $\geq k$ containing x?

Explain why INDN belongs to the complexity class NP.

ii) Show that INDN is NP-complete.

You may assume that the following problem IND is NP-complete: given a graph G, does G have an independent set of size $\geq k$?

The four parts carry, respectively, 30%, 20%, 15%, and 35% of the marks.