Exam : Algorithms and Complexity (COMP90038_2020_SM2)

Started: Nov 23 at 10:00

Quiz Instructions

Academic Integrity Declaration

By commencing and/or submitting this assessment I agree that I have read and understood the **University's policy on academic integrity. (https://academicintegrity.unimelb.edu.au/#online-exams)**

I also agree that:

1. Unless paragraph 2 applies, the work I submit will be original and solely my own work (cheating);
2. I will not seek or receive any assistance from any other person (collusion) except where the work is for a designated collaborative task, in which case the individual contributions will be indicated; and,
3. I will not use any sources without proper acknowledgment or referencing (plagiarism).
4. Where the work I submit is a computer program or code, I will ensure that:
   a. any code I have copied is clearly noted by identifying the source of that code at the start of the program or in a header file or, that comments inline identify the start and end of the copied code; and
   b. any modifications to code sourced from elsewhere will be commented upon to show the nature of the modification.

Short Answer Questions

Question 1

4 pts

The following algorithm uses a *divide and conquer* strategy to calculate a quantity over the array *A*[0]..*A*[*n*-1]. It is initially invoked by calling DCCompute(*A*,0,*n*-1).

```
function DCCompute(A,lo,hi)
  if (lo == hi)
    return A[lo]
  else if (lo > hi)
    return 0
  else
    mid = (lo + hi)/2
    a = DCCompute(A,lo,mid)
    b = DCCompute(A,mid+1,hi)
    return COMBINE(a,b)
  endif
```

This algorithm can be made to compute different quantities of the array *A* by choosing different implementations for the function COMBINE(a,b)

For each quantity below, match it with the correct implementation of COMBINE. If none of the implementations of COMBINE correctly compute that quantity, choose "None"

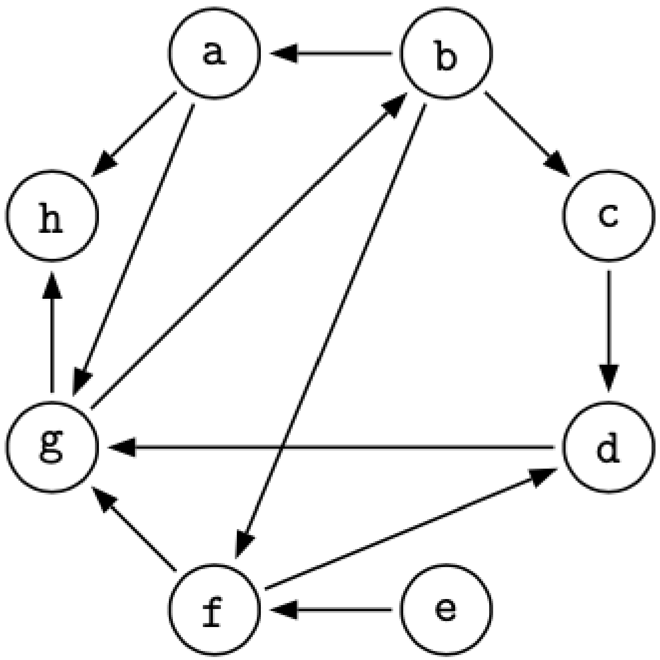
The minimum (smallest) element of A, or 0 if A is empty	COMBINE(a,b) = if a < bth
The sum of the elements of A, or 0 if A is empty	COMBINE(a,b) = a + b

Some positive number, when A contains only positive numbers; otherwise a negative number if A contains at least one non-positive number (i.e. a number <= 0); or 0 if A is empty	COMBINE(a,b) = if a > 0 ar
The number of elements in A	None

Question 2

4 pts

Consider the following directed graph.



For the following questions, assume that ties are resolved by taking nodes in alphabetical order.

Suppose we perform a depth-first traversal (DFS) on this graph, starting from node 'a'. Which node will be the **fourth** node to be visited: [ ]. Which node will be visited **second to last**: [ ]

Suppose we perform a breadth-first traversal (BFS) on this graph, starting from node 'a'. Which node will be visited **last**: [ ]. Which node will be the **fifth** node that is visited: [ ]

Question 3

4 pts

Suppose you have written three divide-and-conquer algorithms, called A_1, A_2, and A_3, to process an array of n elements. The recurrence relations for the complexity of each of the algorithms is given below, where T_i is the recurrence relation for the complexity of algorithm A_i.

$T_1(n) = 9T_1(n/3) + 5n^2$

$T_2(n) = 2T_2(n/2) + 5n$

$T_3(n) = 4T_3(n/3) + 4n^2$

Using the Master Theorem (provided below), or otherwise, determine whether each of the following statements is correct. **Select all true statements**.

Master Theorem

Master Theorem

If $f(n) \in \Theta(n^d)$ where $d \geq 0$, then, given the recurrence

$$T(n) = aT(n/b) + f(n)$$

we have that:

$$T(n) = \begin{cases} \Theta(n^d) & \text{if } a < b^d \\ \Theta(n^d \log n) & \text{if } a = b^d \\ \Theta(n^{\log_b a}) & \text{if } a > b^d \end{cases}$$

- [ ] Algorithm A2 has equal complexity to algorithm A3
- [ ] Algorithm A1 has equal complexity to algorithm A2

- [x] Algorithm A2 has lower complexity than algorithm A3
- [ ] Algorithm A1 has lower complexity than algorithm A3
- [ ] Algorithm A1 has equal complexity to algorithm A3
- [ ] Algorithm A1 has lower complexity than algorithm A2

Question 4

1 pts

Suppose

$f(n) = 4n^2$

$g(n) = n + 2n^3$

Choose the option below that *most precisely* describes the relationship between f(n) and g(n).

- ( ) $f(n) \in \Omega(g(n))$
- (•) $f(n) \in O(g(n))$
- ( ) $f(n) \in \Theta(g(n))$

Question 5

1 pts

Suppose

$f(n) = 3^{n+5}$

$g(n) = 3^n$

Choose the option below that *most precisely* describes the relationship between f(n) and g(n).

- ○ $f(n) \in O(g(n))$
- ○ $f(n) \in \Omega(g(n))$
- ◉ $f(n) \in \Theta(g(n))$

Question 6

1 pts

Suppose

$f(n) = 32n + 5 + 16^n$

$g(n) = 81n^2 + 16 + 4^{2n}$

Choose the option below that *most precisely* describes the relationship between f(n) and g(n).

- ○ $f(n) \in \Theta(g(n))$
- ○ $f(n) \in \Omega(g(n))$
- ◉ $f(n) \in O(g(n))$

Question 7

1 pts

Suppose

$f(n) = 2^n$

$g(n) = 2^{5n}$

Choose the option below that *most precisely* describes the relationship between f(n) and g(n).

- ( ) $f(n) \in \Omega(g(n))$
- (•) $f(n) \in O(g(n))$
- ( ) $f(n) \in \Theta(g(n))$

Question 8

1 pts

Suppose

$f(n) = n^{0.5}$

$g(n) = 5 \log n$

Choose the option below that *most precisely* describes the relationship between f(n) and g(n).

○ f(n) ∈ Θ(g(n))

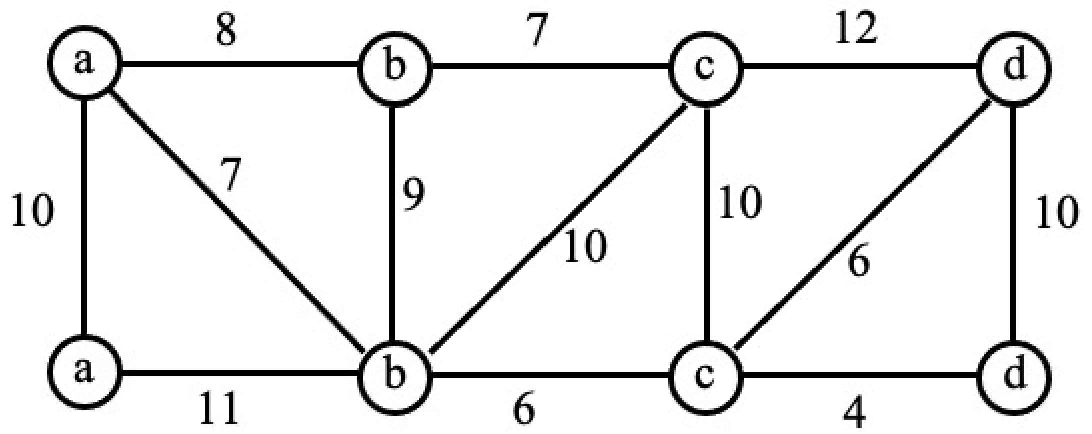
○ f(n) ∈ O(g(n))

◉ f(n) ∈ Ω(g(n))

Question 9

4 pts

Consider the weighted undirected graph with eight nodes shown below:



(a) In the tables below, write down the shortest distance from node 'a' to each of the nodes in the graph, when Dijkstra's algorithm is run on the graph.

node a to a: 0, node a to b:

node a to d: 27, node a to (

node a to g: 13, node a to I

(b) The resulting tree is also the minimum spanning tree? Answer TRUE/FALSE:

FALSE

Question 10

4 pts

Use Warshall's algorithm to compute the transitive closure of the graph with the following adjacency matrix:

0	0	1	1
0	0	0	0
1	0	0	1
1	1	0	0

Choose the correct values for the unknown variables after each of the four steps for the algorithm.

After the first step:

a_1	0	1	1
0	0	0	0
1	b_1	1	1
1	1	c_1	1

a1=0, b1=0, c1=1

After the second step:

a_2	0	1	1
0	0	0	0
1	b_2	1	1
1	1	c_2	1

a2=0, b2=0, c2=1

After the third step:

a_3	0	1	1
0	0	0	0
1	b_3	1	1
1	1	c_3	1

a3=1, b3=0, c3=1

After the fourth and final step:

a_4	1	1	1
0	0	0	0
1	b_4	1	1
1	1	c_4	1

a4=1, b4=1, c4=1

Question 11

4 pts

For the input $[44,\ 51,\ 70,\ 41,\ 26,\ 85,\ 39,\ 56]$ and hash functions $h(k) = k \mod 13$ and $h'(k) = 3 + k \mod 7$, construct the closed hash table that results from inserting the items in the given order, using double hashing.

0	1	2	3	4	5	6	7	8	9	10	11	12

Enter your answers into the three tables:

0	1	2	3

26, 0, 41, 0

4	5	6	7

56, 44, 0, 85

8	9	10	11	12

39, 0, 0, 0, 51

Question 12

4 pts

Consider the pattern $SAUR$ (of length 4) and the text

$TYRANNOSAURUS$

(a) How many character comparisons will the brute-force string matching algorithm make before locating the pattern in the text?

11

(b) How many character comparisons will Horspool's algorithm make before locating the pattern in the text?

7

Question 13

3 pts

Consider the following array $H = [70,\ 10,\ 40,\ 89,\ 68,\ 91,\ 60]$. Using the array representation:

(a) Construct a max-heap bottom up

[91, 89, 70, 68, 10, 40, 60]

(b) Remove the maximum element

[60, 70, 89, 10, 68, 40, 91]

(c) RE-heapify H

[89, 70, 60, 10, 68, 40]

Algorithm Design Questions

Question 14

8 pts

Suppose that A is an array A[0]...A[n-1] of n two-dimensional points. That is, each element A[i] is a point (x_i,y_i). Suppose also that the array is sorted in ascending order according to the points' y-coordinates. Each consecutive pair of points (x_i,y_i) -- (x_{i+1},y_{i+1}) defines a line segment for which, since the array is sorted by y-coordinates, $y_{i+1} \geq y_i$.

Design an algorithm to compute the slope of the line segment that crosses the x-axis.

For example, for the array A containing the 4 points:

(1, -5), (-3, -3), (-1, 1), (10, 3)

the line segment that crosses the x-axis is (-3,-3) -- (-1,1). Hence, for this array your algorithm should return 2 (since (1 - (-3)) / ((-1) - (-3)) = 4 / 2 = 2).

Your algorithm is allowed to assume that such a line segment exists with a well-defined slope, and that the number of points $n > 1$. Your algorithm can also assume that no point in A lies on the x-axis (i.e. that for all points (x_i,y_i) in A, $y_i \neq 0$).

Complexity: Full marks will be given for a correct algorithm that runs in time O(log n); half marks for a correct algorithm that is less efficient.

Edit View Insert Format Tools Table

17px Paragraph B I U A T²

p

117 words </>

Question 15

6 pts

In this question we will consider a prototype robot that the Robotics team in the Department of Mechanical Engineering are building. This particular prototype is in the testing phase for two legged walking as a form of locomotion. This prototype is able to take steps with varying lengths. In this question we will focus on the step lengths: 1, 2, 3 and 4 units. We will need to find an efficient method to calculate the total number of ways the robot can cover a distance of n units given these four possible step sizes.

For example, when n = 3 units, there are 4 possible ways this prototype robot could cover this distance:
1 unit step + 1 unit step + 1 unit step
1 unit step + 2 unit step
2 unit step + 1 unit step
3 unit step

When n = 4 units, there are 8 possible ways this prototype robot could cover this distance:
1 unit step + 1 unit step + 1 unit step + 1 unit step
1 unit step + 2 unit step + 1 unit step
2 unit step + 1 unit step + 1 unit step
1 unit step + 1 unit step + 2 unit step
2 unit step + 2 unit step
1 unit step + 3 unit step
3 unit step + 1 unit step
4 unit step

Write a dynamic programming algorithm that computes the total number of ways a distance of n units could be covered by the prototype robot taking steps sizes of 1, 2, 3 and 4 units. For full marks your algorithm must run in $O(n)$ time. Your algorithm must be written in correct, unambiguous and appropriately commented pseudocode. [6 marks].

Edit View Insert Format Tools Table

17px Paragraph B I U A T²

p 0 words </>

Saved at 12:12 Submit Quiz