Past Exam 3

Faculty of Information Technology

EXAM CODES:	FIT2004
TITLE OF PAPER:	Algorithms and data structures
EXAM DURATION:	2 hours 10 mins

Rules

During your eExam, you must not have in your possession any item/material that has not been authorised for your exam. This includes books, notes, paper, electronic device/s, smart watch/device, or writing on any part of your body. Authorised items are listed above. Items/materials on your device, desk, chair, in your clothing or otherwise on your person will be deemed to be in your possession. Mobile phones must be switched off and placed face-down on your desk during your exam attempt.

You must not retain, copy, memorise or note down any exam content for personal use or to share with any other person by any means during or following your exam. You are not allowed to copy/paste text to or from external sources unless this has been authorised by your Chief Examiner.

You must comply with any instructions given to you by Monash exam staff.

As a student, and under Monash University's Student Academic Integrity procedure, you must undertake all your assessments with honesty and integrity. You must not allow anyone else to do work for you and you must not do any work for others. You must not contact, or attempt to contact, another person in an attempt to gain unfair advantage during your assessment. Assessors may take reasonable steps to check that your work displays the expected standards of academic integrity.

Failure to comply with the above instructions, or attempting to cheat or cheating in an assessment may constitute a breach of instructions under regulation 23 of the Monash University (Academic Board) Regulations or may constitute an act of academic misconduct under Part 7 of the Monash University (Council) Regulations.

<u>Authorised Materials</u>

CALCULATORS	☐ YES	☑ NO
DICTIONARIES	☐ YES	☑ NO
NOTES	☐ YES	☑ NO
WORKING SHEETS	☑ YES	☐ NO
PERMITTED ITEM	☐ YES	☑ NO

if yes, items permitted are:

Instructions

- This is a **closed book** exam with **Specifically permitted items**.
- Please answer **ALL** questions.
- In this assessment, you must **not** use generative artificial intelligence (AI) to generate any materials or content in relation to the assessment task.
- Once the exam duration is finished, your exam will automatically submit. Please ensure you finalise your answers before the end of the allocated exam time.

Instructions

Information

You can review your exam instructions by clicking the 'Show Instructions' button above.

Analysis of Algorithms: Correctness and Complexity

Question 1

For constants b and c, consider the recurrence relation given by:

3
Marks

- $T(n) = b$, if $n=1$
- $T(n) = 2 * T(n/4) + c * n$, if $n>1$

Which of the following statements is **TRUE**?

Select one:

- ○ a.
  $T(n) = \Theta(n)$
- ○ b.
  $T(n) = \Theta(n * \log n)$
- ○ c.
  $T(n) = \Theta(1)$
- ○ d.
  $T(n) = \Theta(n^{1/2})$
- ○ e.
  $T(n) = \Theta(\log n)$

Question 2

Consider the following algorithm for counting the number of elements with a factor m in a list L.

2
Marks

```
1.  function myFunc(L[1...n], m):
2.      x = 0
3.      i = 1
4.      while i <= n:
5.          # loop invariant here
6.          if L[i] % m == 0:
7.              x = x + 1
8.          else:
9.              x = x + 0
10.         i = i + 1
11.     return x
```

The loop invariant for this algorithm is: *x* is the number of items with a factor of *m* in list *L*[1...i-1].

Which of the following is **TRUE**?

Select one or more:

☐ a.
The loop invariant for the algorithm shown should be: *x* is the number of items with a factor of *m* in list *L*[1...i].

☐ b.
The invariant at the point where i=n+1 implies correctness of the algorithm.

☐ c.
The invariant at the point where i=n implies correctness of the algorithm.

☐ d.
Does the following invariant hold if placed just before line 10: x is the number of items with factor of m in list L[1...i]?

Question 3

Describe how to sort *N* integers in the range 0 to $N^5 - 1$ in O(N) time. **Justify** how your solution meets the time complexity.

2
Marks

Question 4

For each of the following operations, **determine its worst-case big-Θ complexity.**

2 Marks

In this question,

- The graph *G* is a directed weighted graph.
- *V* refers to the number of vertices in the graph.
- *E* refers to the number of edges in the graph.
- *N(A)* refers to the number of neighbors of vertex *A*.

Assume that in the adjacency list representation, the interior list is unsorted.

Time complexity to obtain the total weight sum of incoming edges to vertex A in an adjacency matrix representation.	· Θ(V^3) · Θ(log V) · Θ(E^3) · Θ(V^2) · Θ(E^2) · Θ(log E) · Θ(E log V) · Θ(V) · Θ(1)
Time complexity to determine if there is a directed edge from vertex A to vertex B in an adjacency matrix representation.	· Θ(V^3) · Θ(log V) · Θ(E^3) · Θ(V^2) · Θ(E^2) · Θ(log E) · Θ(E log V) · Θ(V) · Θ(1)
Time complexity to run Breadth-First Search from vertex A in an adjacency matrix representation.	· Θ(V^3) · Θ(log V) · Θ(E^3) · Θ(V^2) · Θ(E^2) · Θ(log E) · Θ(E log V) · Θ(V) · Θ(1)
Time complexity to determine if there is a directed edge from vertex A to vertex B; and also vertex B to vertex A in an adjacency list representation.	· Θ(V^3) · Θ(log V) · Θ(E^3) · Θ(V^2) · Θ(E^2) · Θ(log E) · Θ(E log V) · Θ(V) · Θ(1)

Question 5

For each of the statements below, determine if the statement is **TRUE** or **FALSE**.

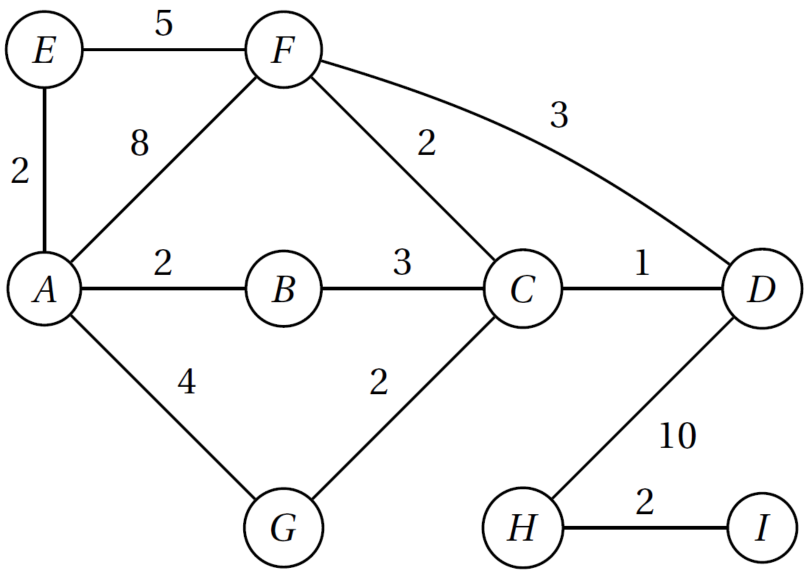
2 Marks

Given an adjacency-list representation of a directed graph G = (V, E), it takes Θ(V+E) to compute the in-degree of every vertex.	· TRUE · FALSE
Suppose that we are given a weighted directed graph G = (V, E) in which outgoing edges from the source vertex s may have negative weights, all other edge weights are non-negative, and there are no negative-weight cycles. You can run Dijkstra’s algorithm to find the correct shortest paths from vertex s in this graph.	· TRUE · FALSE
Suppose that T is a tree constructed by running Dijkstra’s algorithm on a weighted connected graph G, it must be true that T is a minimum spanning tree of G.	· TRUE · FALSE

Question 6

Consider the weighted undirected graph below.

1
Mark



What is the height of a tree rooted at vertex **A** using Breadth First Search? Just type the numerical answer.

Question 7

Which of the following statements are **TRUE**?

2
Marks

Select one or more:

☐ a.
Bellman-Ford can sometimes terminate earlier without running the outer loop V times if there is no negative cycle in the directed weighted graph.

☐ b.
Bellman-Ford would require $\Theta(V^2)$ auxiliary space for computation.

☐ c.
Floyd-Warshall can sometimes terminate earlier without running the outer loop V times if there is no negative cycle in the directed weighted graph.

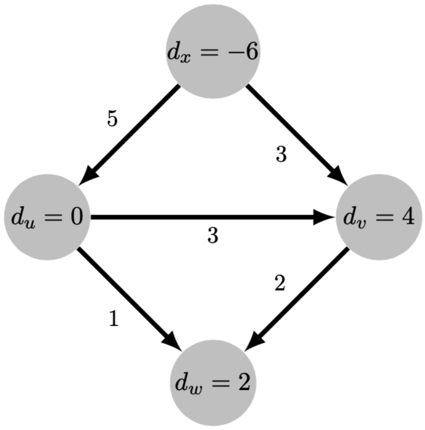
☐ d.
It is not possible for the diagonal values in the Floyd-Warshall memo-matrix to have a negative value.

Question 8

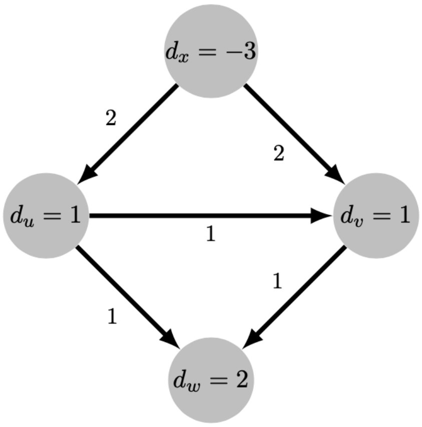
Consider the following two problems of circulation with demands, in which the demands are indicated in each vertex, and the capacity in each edge.

4
Marks

Problem 1:



Problem 2:



Which of those problems have feasible solutions?

Select one:

- a. Only Problem 1 has a feasible solution.
- b. Only Problem 2 has a feasible solution.
- c. Both Problem 1 and Problem 2 have feasible solutions.
- d. Neither Problem 1 nor Problem 2 has a feasible solution.

Question 9

Which of the following statements are **TRUE**?

2
Marks

Select one or more:

☐
a.
If the graph have negative edges, it is not possible to obtain a minimum-spanning tree using Kruskal's algorithm.

☐
b.
It is possible to obtain a maximum-spanning tree using Kruskal's algorithm if the edges are processed in descending order.

☐
c.
Given a connected weighted undirected graph, the minimum-spanning tree of that graph is always unique.

☐
d.
A minimum-spanning tree obtained using Prim's algorithm is unique if the weight of all edges are unique, even if a different starting vertex is chosen.

Question 10

Show that the complexity of the 0/1 knapsack problem is actually exponential.

2
Marks

Question 11

Select the **WRONG or FALSE** statement(s) about the dynamic programming algorithms.

2
Marks

Select one or more:

☐
a.
Writing a dynamic programming algorithm, using a bottom-up approach is asymptotically faster than top-down approach.

☐
b.
The running time of a dynamic programming algorithm is always $\Theta(n)$ where n is the number of subproblems.

Question 12

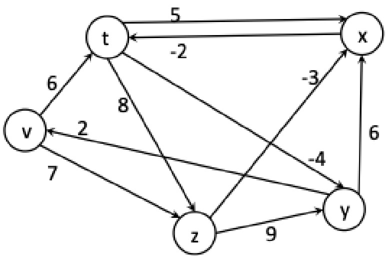
Consider Bellman-Ford algorithm

3
Marks

Algorithm Bellman-Ford

```
function BELLMAN_FORD(G = (V, E), s)
    dist[1..n] = ∞
    pred[1..n] = null
    dist[s] = 0
    for k = 1 to n − 1 do
        for each edge e in E do
            RELAX(e)
    return dist[1..n], pred[1..n]
```

and the following weighted directed graph.



Let y be the source node for the execution of the Bellman-Ford algorithm.

If the edges are relaxed in the following order (v, t), (x, t), (t, z), (v, z), (t, y), (z, y), (y, x), (z, x), (t, x), (y, v).

Run the outer loop of the algorithm for two iterations and then **answer** the following questions.

What is the distance value of vertex t after the second iteration of the outer loop is done?	· t · 3 · 4 · 0 · 2 · 8 · z · x · y · 9 · 6 · v
What is the distance value of vertex z after the second iteration of the outer loop is done?	· t · 3 · 4 · 0 · 2 · 8 · z · x · y · 9 · 6 · v
What is the distance value of vertex x after the second iteration of the outer loop is done?	· t · 3 · 4 · 0 · 2 · 8 · z · x · y · 9 · 6 · v
What is the predecessor vertex of vertex t after the second iteration of the outer loop is done?	· t · 3 · 4 · 0 · 2 · 8 · z · x · y · 9 · 6 · v
What is the predecessor vertex of vertex z after the second iteration of the outer loop is done?	· t · 3 · 4 · 0 · 2 · 8 · z · x · y · 9 · 6 · v
What is the predecessor vertex of vertex v after the second iteration of the outer loop is done?	· t · 3 · 4 · 0 · 2 · 8 · z · x · y · 9 · 6 · v

Question 13

Which of the following statements are true regarding probing techniques for hash tables? Please **select only the correct answers**.

2
Marks

Select one or more:

☐ a.

Linear probing avoids primary clustering.

☐ b.

Quadratic probing causes secondary clustering.

☐ c.

Quadratic probing causes primary clustering.

☐ d.

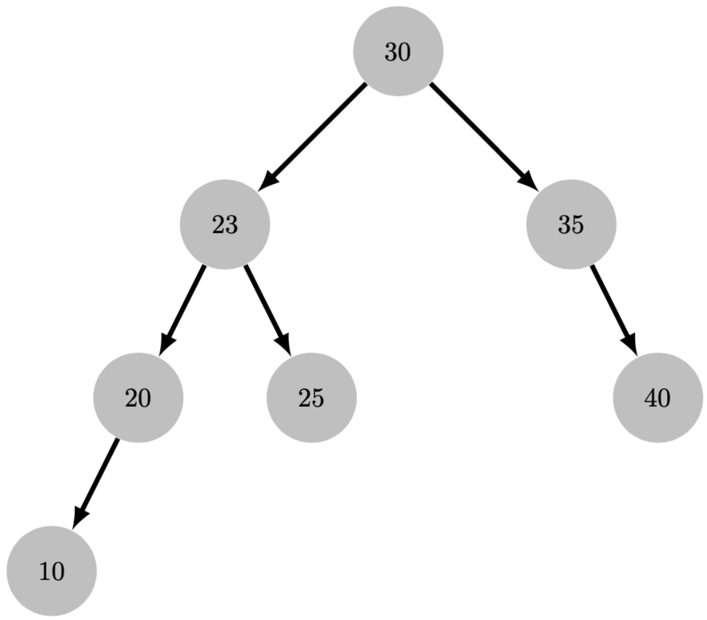
Quadratic probing can fail to insert an element even if there is still some empty locations in the hash table.

☐ e.

Linear probing can fail to insert an element even if there is still some empty locations in the hash table.

Information

For the **next three questions** consider that you initially have the following AVL tree



and then perform the following operations in order:

- Delete 25
- Insert 27
- Delete 40

Question 14

After performing the operations, what is the value in the root node of the AVL tree?

1
Mark

Select one:

- a. 23
- b. 27
- c. 30
- d. 35
- e. 20
- f. 10

Question 15

After performing the operations, what is the value of left child of the node 30?

1
Mark

Select one:

- a. 23
- b. 20
- c. 27
- d. Node 30 has no left child.
- e. 10
- f. 35

Question 16

After performing the operations, what is the value of left child of the node 23?

1
Mark

Select one:

- a. 35
- b. Node 23 has no left child.
- c. 27
- d. 30
- e. 10
- f. 20

Question 17

Assume that we are constructing the suffix array for a string S using the prefix doubling approach.

2
Marks

We have already sorted the suffixes for string S according to their first 1 character(s); with the corresponding rank array shown below:

ID	1	2	3	4	5	6	7	8	9	10	11
Rank	11	2	6	2	6	2	10	9	6	2	1

We are now sorting on the first 2 character(s), comparing the suffixes on their first 2 characters in O(1) using prefix doubling.

Which of the following statements are **TRUE**?

Select one or more:

☐ a.
Suffixes with ID-4 and ID-10 will have a different rank after sorting the first 2 characters where suffix ID-10 will have a smaller rank than suffix ID-4.

☐ b.
Suffixes with ID-4 and ID-6 still have the same rank after sorting the first 2 characters.

☐ c.
Suffixes with ID-3 and ID-9 still have the same rank after sorting the first 2 characters.

☐ d.
Suffixes with ID-2 and ID-4 will have a different rank after sorting the first 2 characters where suffix ID-4 will have a smaller rank than suffix ID-2.

Applications

Question 18

Two paths in a graph are edge disjoint if they have no edges in common.

Given a directed graph G with E edges and V vertices, we would like to determine the maximum number of edge-disjoint paths from vertex *s* to vertex *t* with an algorithm that runs in time complexity O(E*V). **Describe how** to determine the maximum number of edge-disjoint *s*–*t* paths; **justify** why it works and why your solution is within the time complexity.

5
Marks

Question 19

Arbitrage is the process of exploiting conversion rates between commodities to make a profit. Arbitrage may occur over many steps. For example, one could purchase US Dollars using Australian Dollars, convert it into Great British Pounds, and then back into Australian Dollars. If the prices or exchange rates were right, this could result in a profit. Given a list of currencies and the best conversion rate between each pair, **devise an algorithm** that determines whether arbitrage is possible, i.e. whether or not you could make a profit. Your algorithm should run in $O(N^3)$ where N is the number of currencies.

3
Marks

Question 20

Devise an algorithm that given a box with N different locks and N corresponding keys, matches the keys and locks in average-case time complexity $O(N \log N)$. Each lock matches only one key, and each key matches only one lock. You can try a key in a lock to determine whether the key is larger than, smaller than or fits the lock. However, you cannot compare two keys or two locks directly.

3
Marks

Question 21

You are running a tennis tournament for your local community.

3 Marks

- There are a total of *N* participants.
- Each participants have their own Elo rating in the range of 0 to 3000.
- The Elo rating for each of the participants are stored in a list $[P_1, P_2, P_3, ..., P_N]$ where P_i is the Elo rating for participant *i*. This list is not sorted.

You want to break the tournament down into 2 categories:

- The professional level where the top-20% in Elo ratings would participate in.
- The amateur level where the remaining bottom-80% in Elo ratings would participate in.

For each of the categories:

- The top-16 highest rated participants will automatically be qualified to the double-elimination playoff brackets.
- The remaining participants will need to participate in the group stage to qualify for the playoff.

Describe an efficient algorithm to group the participants into their respective categories and tournament stages.

Question 22

You are overseeing an intergalactic defense force, that protects your galaxy from otherworldly threats.

2 Marks

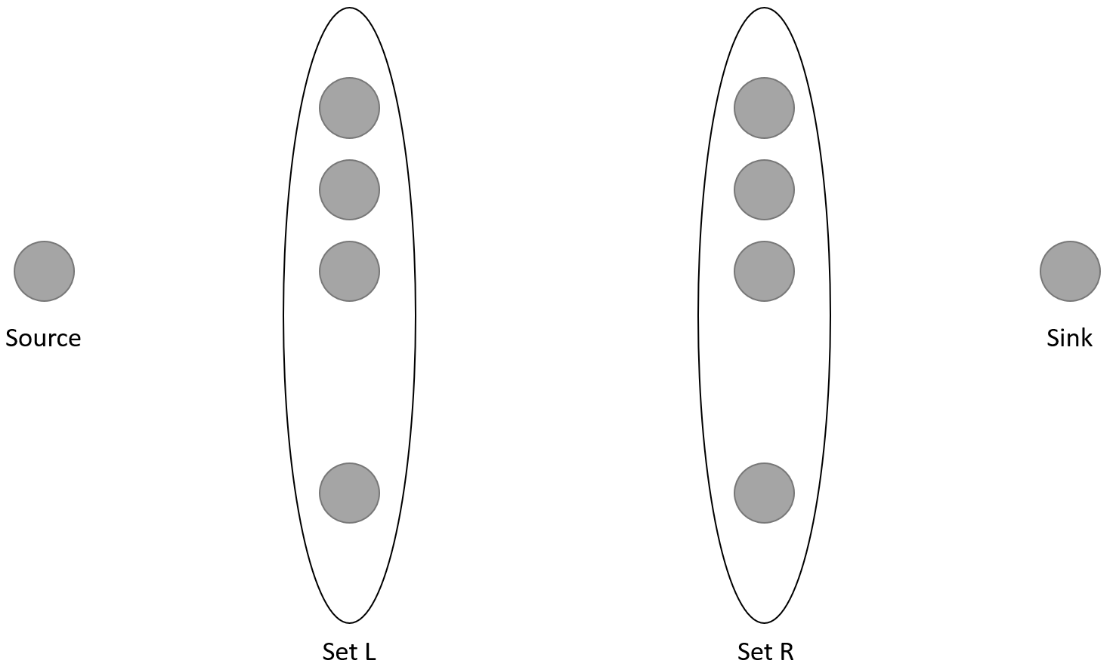
- There are 300 planets in your galaxy.
- There are 200 superheroes in your defense force.

When enemies attacks, each planet can send a request for help to superheroes.

- Each planet can send up to a maximum of 10 requests for help.
- Each planet would require only one superhero to defend the planet.

However, a superhero can only defend up to 2 planets at the same time. Thus, it is a dilemma for you to assign the superheroes to their duties. You would want to defend as many planets as possible!

As a commander with a Bachelors degree in Computer Science, you know you can model this problem as a maximum flow problem for bipartite matching as shown below:



Which of these models are the **most suitable**?

Select one:

◯ a.

- Set L: planet
- Set R: superheroes
- Capacity of each edge from source to each node of set L: 300
- Capacity of each edge from each node of set L to each node of set R: 10
- Capacity of each edge from each node of set R to sink: 200

◯ b.

- Set L: superheroes
- Set R: planets
- Capacity of each edge from source to each node of set L: 2
- Capacity of each edge from each node of set L to each node of set R: 1
- Capacity of each edge from each node of set R to Sink: 10

◯ c.

- Set L: superheroes
- Set R: planets
- Capacity of each edge from source to each node of set L: 2
- Capacity of each edge from each node of set L to each node of set R: 1
- Capacity of each edge from each node of set R to sink: 1

◯ d.

- Set L: superheroes
- Set R: planets
- Capacity of each edge from source to each node of set L: 200
- Capacity of each edge from each node of set L to each node of set R: 10
- Capacity of each edge from each node of set R to sink: 300