Past Exam 2

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.

Authorised Materials

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.
- 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

3 Marks

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

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

Which of the following statements is true?

Select one:

a. $T(n) = \Theta(n * \log n * \log n)$

b. $T(n) = \Theta(n^2 * \log n * \log n)$

c. $T(n) = \Theta(n * \log n)$

d. $T(n) = \Theta(n^2)$

e. $T(n) = \Theta(n^2 * \log n)$

Question 2

3 Marks

The pseudocode below finds the maximum degree of a graph *G = (V,E)*.

```
function max_degree(G = (V,E)):
    degrees[1..n] = 0
    for each vertex u in V:
        for each edge (u,v):
            degrees[u] += 1
    return max(degrees)
```

Assuming an adjacency list representation, what is the worst-case time complexity, total space complexity, and auxiliary space complexity of this pseudocode in terms of *V* and *E*? Justify your solution.

Question 3

Consider the following algorithm, which returns the sum of the numbers with a factor *m* in the list *L* with *n* items.

1 Mark

```
def myfunc(L[1...n], m):
    x = 0
    loop i from 1 to n:
        # loop invariant here
        if L[i] % m == 0:
            x = x + L[i]
    return x
```

What is an useful loop invariant for this algorithm?

Select one:

- a. x is the sum of all numbers in list L[1...i] % m
- b. x is the sum of all numbers in list L[1...i-1] % m
- c. x is the sum of all numbers with factor m in list L[1...i-1]
- d. x is the sum of all numbers with factor m in list L[1...n]
- e. x is the sum of all numbers in list L[1...n] % m
- f. x is the sum of all numbers with factor m in list L[1...i]

Question 4

Justify that the loop invariant you have chosen is true each time the loop runs and when it terminates.

2 Marks

Question 5

Recall that the **Radix Sort** algorithm covered in lectures works as follows, for sorting an array of integers:

3 Marks

- Sort the array one digit at a time, from least significant (rightmost) digit to most significant (leftmost) digit.
- For each digit, sort using the stable version of counting sort.

The integers in the array do not have to be represented in base-10. Is it a good idea to increase the base representation to be as high as possible? Why, or why not? Explain your reasoning.

Question 6

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

2
Marks

In this question,

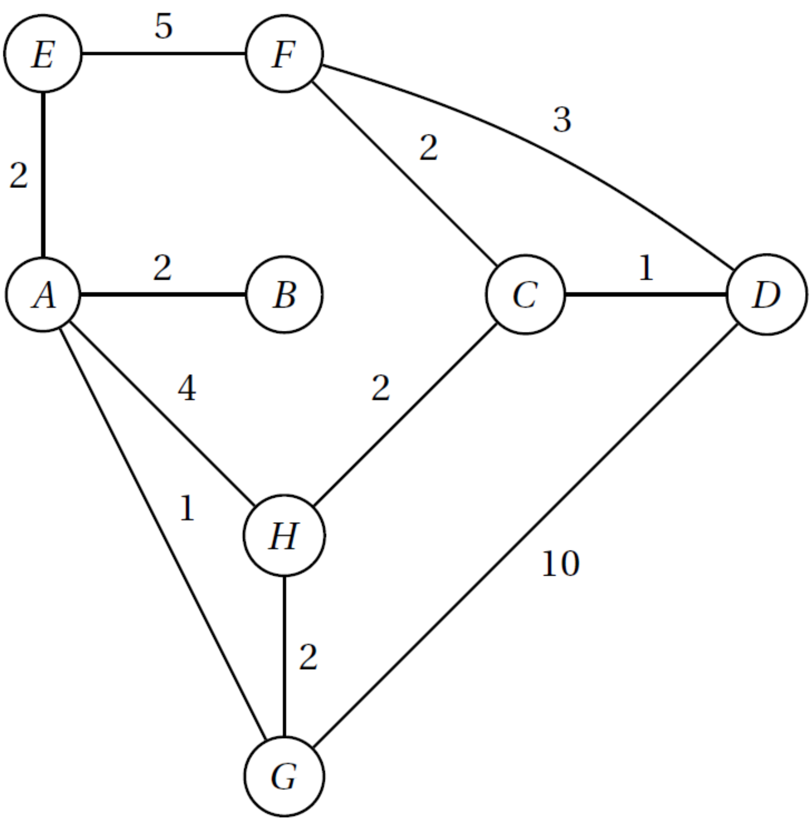
- The graph G is a connected directed weighted graph.
- *V* refers to the number of vertices in the graph.
- *E* refers to the number of edges in the graph.
- *N(A)* refers to the number of neighbors of vertex A.

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

Time complexity to determine if vertex B is adjacent to vertex A in an adjacency matrix representation.	· V · N(A) · log E · 1 · log V · V^2 · E
Time complexity to obtain all incoming edges for vertex A in an adjacency list representation.	· V · N(A) · log E · 1 · log V · V^2 · E
Time complexity to determine if there is an edge from vertex A to vertex B in an adjacency list representation.	· V · N(A) · log E · 1 · log V · V^2 · E
Time complexity to print all outgoing edges for vertex A in an adjacency list representation.	· V · N(A) · log E · 1 · log V · V^2 · E

Information

Consider the **weighted undirected graph** below and answer the following questions.



Question 7

Perform a breadth-first search on the graph given above, starting from A.

Whenever you have a choice between 2 vertices, break ties in ascending alphabetical order.

2
Marks

1st visited vertex	· A · G · C · E · D · H · F · B
2nd visited vertex	· A · G · C · E · D · H · F · B
3rd visited vertex	· A · G · C · E · D · H · F · B
4th visited vertex	· A · G · C · E · D · H · F · B
5th visited vertex	· A · G · C · E · D · H · F · B
6th visited vertex	· A · G · C · E · D · H · F · B
7th visited vertex	· A · G · C · E · D · H · F · B
8th visited vertex	· A · G · C · E · D · H · F · B

Question 8

Which of the following is true?

2
Marks

Select one or more:

☐
a.
Prim's minimum spanning tree algorithm will produce a maximum spanning tree if a max-heap is used instead of a min-heap.

☐
b.
Negating the weight of edges in a graph before running the Kruskal's minimum spanning tree algorithm will indicate the edges for a maximum spanning tree.

☐
c.
The parent-array in the union-find (with union-by-size) data structure is used in the Kruskal's minimum spanning tree algorithm do indicate the edges of the minimum spanning tree.

☐
d.
Prim's algorithm for the minimum spanning tree is a greedy algorithm that will not work if the graph have a negative edge.

Question 9

The following pseudocode is for the generic Depth First Search algorithm covered in lectures.

3 Marks

```
function TRAVERSE(G=(V,E))
    visited[1..n] = false
    for each vertex u = 1 to n do
        if not visited[u] then
            DFS(u)
```

```
function DFS(u)
    visited[u] = true
    for each vertex v adjacent to u do
        if not visited[v] then
            DFS(v)
```

How would you modify this algorithm to perform a**topological sort** of the vertices in a directed acyclic graph? Explain clearly which lines you would add/remove/modify, and what the purpose of each change would be.

Question 10

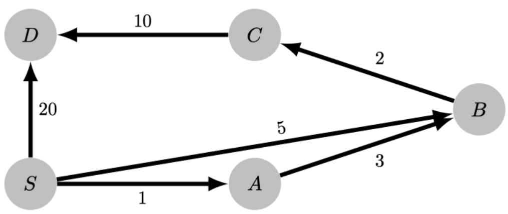
Consider the following version of the Bellman-Ford algorithm

3 Marks

Algorithm 54 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 directed graph



Let S be the source node for the execution of the Bellman-Ford algorithm. If the edges are relaxed in the following order (C, D), (B, C), (A, B), (S, D), (S, B), (S, A), what is the value of dist[A]+dist[B]+dist[C]+dist[D] after the second iteration of the outer loop is finished? Just type the numerical answer.

Question 11

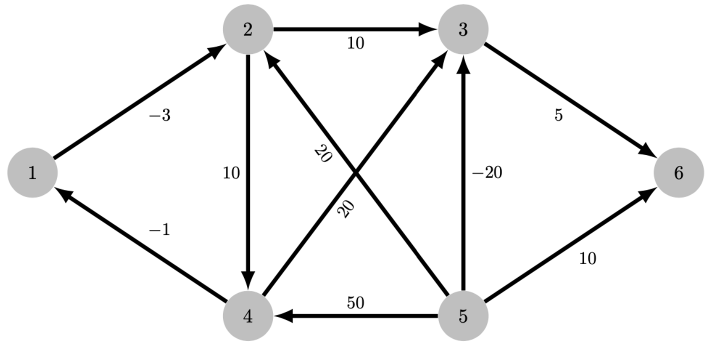
3
Marks

Consider the Floyd-Warshall algorithm

Algorithm 56 Floyd-Warshall

function FLOYD_WARSHALL($G = (V, E)$)
  $dist[1..n][1..n] = \infty$
  $dist[v][v] = 0$ **for all** vertices v
  $dist[u][v] = w(u, v)$ **for all** edges $e = (u, v)$ **in** E
  for each vertex $k = 1$ **to** n **do**
    for each vertex $u = 1$ **to** n **do**
      for each vertex $v = 1$ **to** n **do**
        $dist[u][v] = \min(dist[u][v], dist[u][k] + dist[k][v])$
  return $dist[1..n][1..n]$

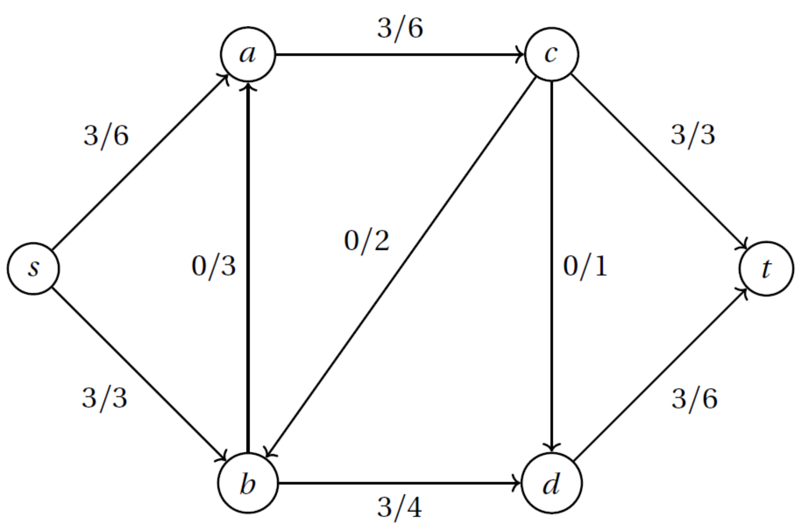
and the following directed graph



After the second iteration of the outer loop of the algorithm is finished,
what is the value of dist[4][3]+dist[5][4]+dist[5][6]? Just type the numerical answer.

Information

Consider the **flow network** below and answer the following questions.



Question 12

What is the **maximum possible flow** for the given flow network above?

Just type in the numerical answer.

2
Marks

Question 13

A cut partitions the vertices into 2 disjoint sets S and T where

- S contains all the vertices on the source side of the cut.
- T contains all the vertices on the sink side of the cut.

Consider the minimum cut of the above flow network.

Select the vertices which are in S from the list of vertices below

1
Mark

Select one or more:

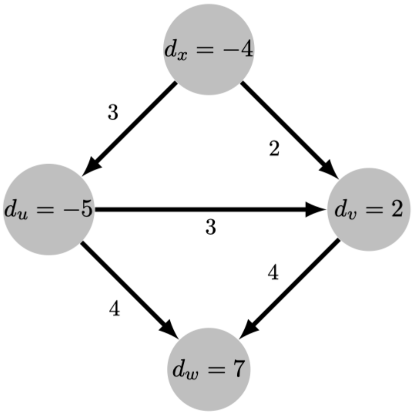
- [ ] t
- [ ] d
- [ ] b
- [ ] s
- [ ] a
- [ ] c

Question 14

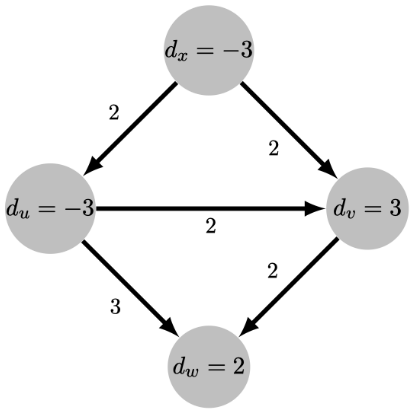
4
Marks

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

Problem 1:



Problem 2:



Which of the problems have feasible solutions?

Select one:

○ a.
Only Problem 1 has a feasible solution.

○ b.
Both Problem 1 and Problem 2 have feasible solutions.

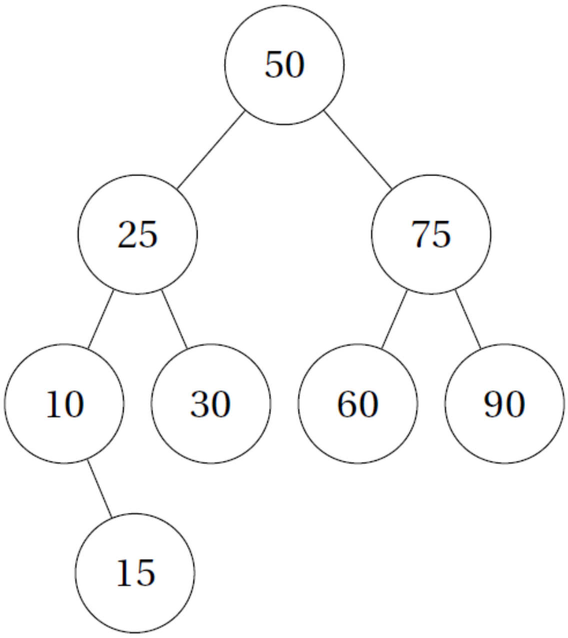
○ c.
Only Problem 2 has a feasible solution.

○ d.
Neither Problem 1 nor Problem 2 has a feasible solution.

Question 15

Consider the following AVL tree below:

2
Marks



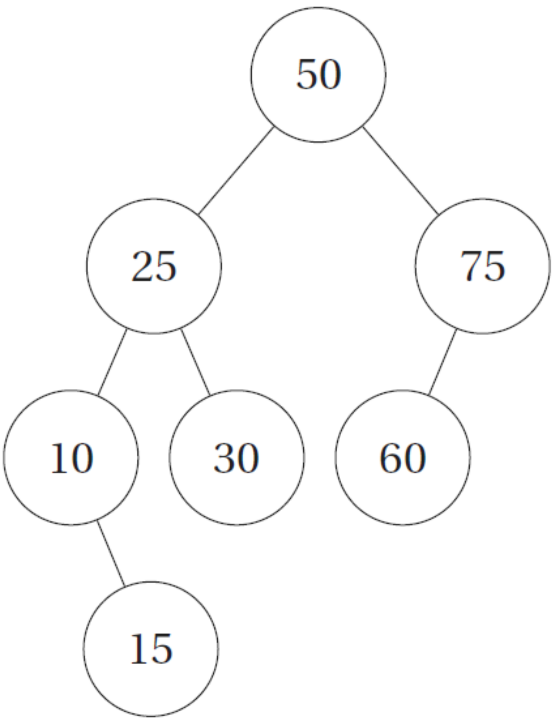
You perform the following operation in order:

- Delete 90.
- Insert 12.

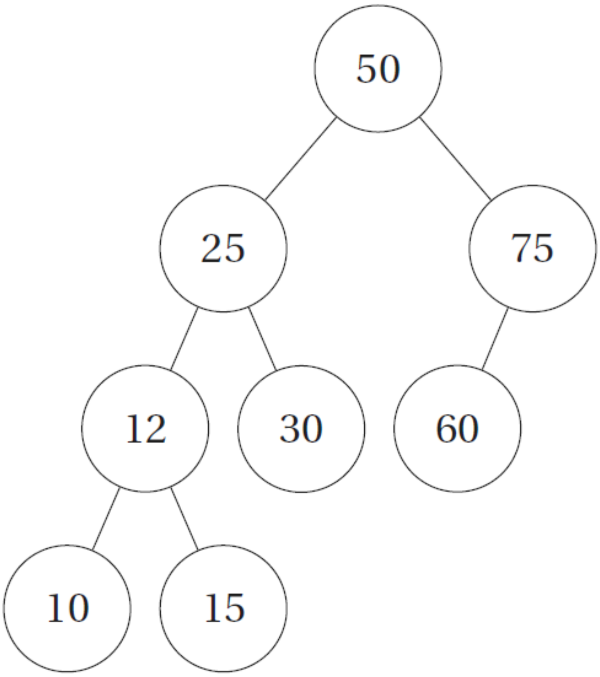
Select the resulting AVL tree after performing the operations above.

Select one:

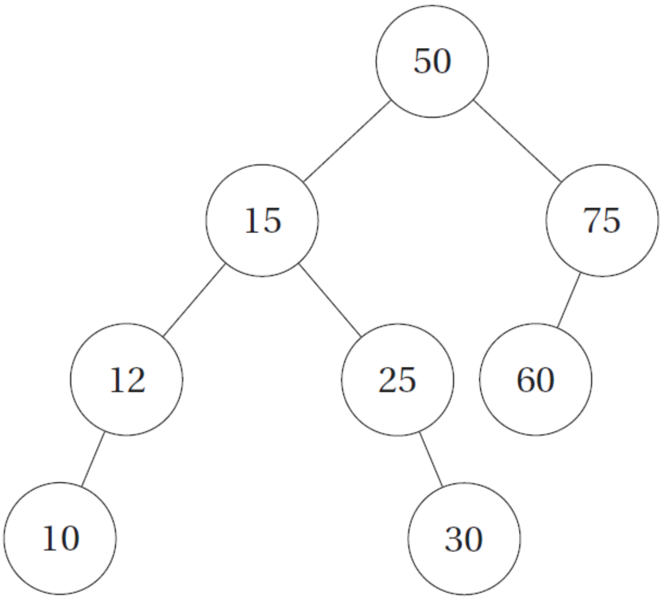
○ a.



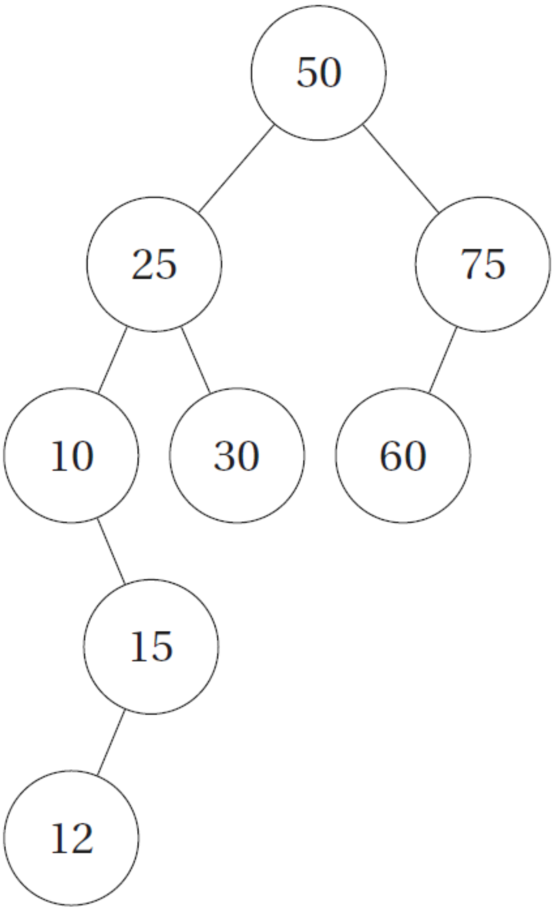
◯ b.



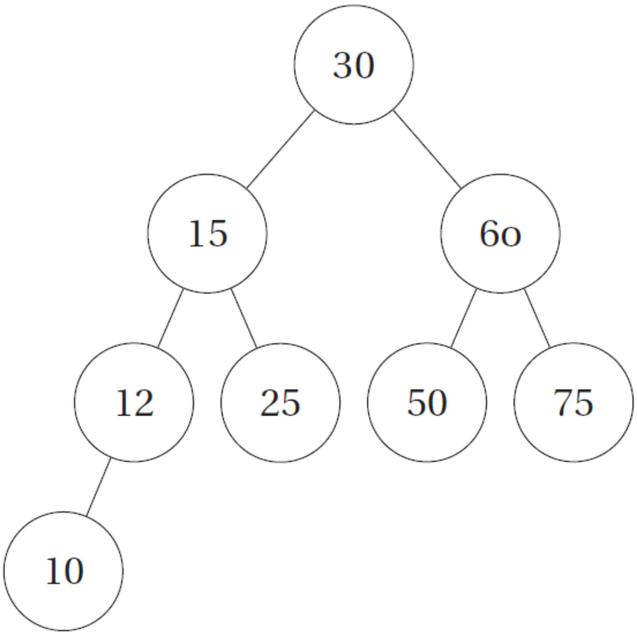
◯ c.



○ d.



○ e.



Question 16

Consider a hash table implemented with separate chaining for collision resolution.

For a hashtable with *N* items, which of the following data structures would cause the worst case time complexity of an insert operation to be Θ(log N) if the data structure is used to keep the separate chains?

2 Marks

Select one or more:

- [ ] a. Binary search tree
- [ ] b. Sorted linked list
- [ ] c. Sorted array
- [ ] d. AVL Tree

Question 17

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

We have already sorted the suffixes for string S according to their first 2 characters; with the corresponding rank array shown below:

2 Marks

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

We are now sorting on the first 4 characters, comparing the suffixes on their first 4 characters in O(1).

Which of the following statements are true?

Select one or more:

- [ ] a. Suffixes with ID4 and ID6 will have a different rank after sorting the first 4 characters where suffix ID6 would have a smaller rank than suffix ID4.
- [ ] b. Suffixes with ID5 and ID7 still have the same rank after sorting the first 4 characters.
- [ ] c. Suffixes with ID4 and ID6 will have a different rank after sorting the first 4 characters where suffix ID4 would have a smaller rank than suffix ID6.
- [ ] d. Suffixes with ID3 and ID5 still have the same rank after sorting the first 4 characters.

Applications

Question 18

3 Marks

You are selecting a team of superheroes to save the world. Unfortunately, you can only bring a fixed amount of them through the portal.

You are given an unsorted list of *N* superheroes where each item is in a tuple of (name, power level). You would want to send:

- The 1st team: the top-10% best superheroes by power-level from that list.
- The 2nd team: the top-10% best superheroes by power-level from the remainder of the list.

You would however not send in the 2nd team until everyone in the 2nd team has a power level greater than or equivalent to the median of the power level of the 1st team. The 2nd team would need to train until they reach that power level.

Describe an efficient algorithm using Quickselect to determine the total power level needed to be gained during training before the 2nd team can be sent out. **Your algorithm should run in *O(N)* time**; and you can assume that you have access to a Quickselect algorithm which runs in *O(N)* time.

Question 19

3 Marks

You are coordinating a industry placement unit.

There are a total of 240 students that are enrolled in the unit; and there are 30 companies to choose from.

- Each student is allowed to select 1 to 3 companies as their preferred placement, but they would only be placed in 1 company at the end.
- Each company would be able to accept 8 students.
- Students would reject any placement that is not in their preferred selection.

You realized that it is not possible for all students to be placed in their preferred company as some companies are more popular than others. You would however try to place as many students in their preferred companies as possible; any students not placed in this round would be placed in the following round.

Describe how you would **model this problem as a maximum flow problem**, which is then **solved using the Ford-Fulkerson method**.

Question 20

You find yourself curiously stranded on a grid (shown in the figure below), unsure of how you got there, or how to leave. Some of the cells of the grid are blocked and cannot be walked through. Anyway, while you're here, you decide to solve the following problem. You are currently standing at the bottom-left corner of the grid, and are only able to move up (to the next row) and to the right (to the next column). You wonder, in how many different ways can you walk to the top-right corner of the grid while avoiding blocked cells? Just type the numerical answer.

4
Marks