MCQs and Fill in the Blanks

1. During Steven's last lecture (Wed, 13 April 2022, 8.00-8.20am), Steven outlined a few things that should **NO LONGER** be used in your C++ code going forward, especially in a big (C++) software engineering project.

One of the 5 options below is an impostor. Select that impostor.

(2 marks)

- do not overuse typedefs, define, or using, e.g.,

  ```
  typedef long long ll;
  #define ll long long
  using ll = long long;
  ```

- do not write any comment in your C++ code
- do not abuse nested ternary operators, e.g.,

  ```
  (t1)?(t2?if_t1truet2true:if_t1truet2false):(if_t1false);
  ```

- do not use

  ```
  #include <bits/stdc++.h>
  ```

  as your only #include
- do not abuse

  ```
  use namespace std;
  ```

2. Steven has a snippet of old C++11 code lying around.

```
for (map<string, int>::iterator it = answer.begin(); it != answer.end(); it++) {
  printf("%s = %d\n", ((string)it->first).c_str(), it->second);
}
```

Which of the option below is the equivalent C++17 (the code compiles and is essentially similar to the old C++11 version)?

(2 marks)

- ```
  for (auto [k, v] : answer)
    printf("%s = %d\n", ((std::string)k).c_str(), v);
  ```
- ```
  for (auto& [k, v] : answer)
    printf("%s = %d\n", ((std::string)k).c_str(), v);
  ```

```
for (auto& [k, v] : answer)
  std::cout << v << " = " << k << std::endl;
```

```
for (auto& [k, v] : answer)
  std::cout << k << " = " << v << std::endl;
```

```
for (auto& [k, v] : answer)
  std::cout << k << " = " << v << '\n';
```

3. Suppose we have an std::multiset (balanced BST) of integers called **ms** that has been pre-populated with a few integers. Notice that there are 8 copies of 7 inside **ms**. Steven just want to delete exactly 1 copy of 7 from **ms** so that after this operation is performed, **ms** contains 7 copies of 7.

```
std::multiset<int> ms = {1,1,5,7,7,7,7,7,7,7,7};
// what should be inserted here to delete just 1 copy of 7 from ms?
for (auto& v : ms)
  std::cout << v << '\n';
```

Please replace the second line in the code snippet above with the correct code.

(2 marks)

```
ms.erase(ms.find(7), ms.end());
```

```
ms.clear();
```

```
ms.erase(ms.find(7));
```

```
ms.removeonecopy(7);
```

```
ms.erase(7);
```

4. What is the time complexity of the function f(n) according to n given the function g() has a time complexity of O(1)?

```
f(int n) {
  for (int i=0;i<n;i++)
    for (int j=0;j<i;j+=2)
      g();
}
```

(2 marks)

- O(n)
- $O(n^2)$
- O(n log n)
- $O(n^3)$
- O(log n)

5. What is the time complexity of the function f(n) according to n given the function g() has a time complexity of O(1)?

```
f(int n) {
  while (n>2) {
    for (int i=0;i<n;i++)
      g();
  n = n / 2;
}
```

(2 marks)

- $O(n^2)$
- $O(n^3)$
- O(n)
- O(n log n)
- O(log n)

6. What is the time complexity of the function f(n) according to n given the function g() has a time complexity of O(1)?

```
f(int n) {
  for (int i=1;i<n;i++)
    for (int j=1;j<n;j+=i)
      g();
}
```

(2 marks)

- $O(n^2)$
- O(n)
- O(log n)
- $O(n^3)$
- O(n log n)

7. According to our lecture, which of the following sorting is not in-place with array implementation?

(2 marks)

- MergeSort
- QuickSort
- All of the algorithms here are all in-place
- BogoSort
- InsertionSort

8. Insertion sort (IS) is better than Selection sort (SS) because

(2 marks)

- IS can handle duplicates and SS cannot
- IS is in-place and SS is not
- SS must always scan all remaining elements for every round
- IS needs recursion and SS doesn't
- IS is stable and SS is not

9. In our usual computer architecture, reading and writing memory takes time. If writing memory takes a lot of time, which sorting algorithm is relatively better in terms of minimizing the number of writes into memory?

(2 marks)

- Bubble Sort
- Cocktail Shaker Sort
- BogoSort
- Selection Sort
- Insertion Sort

10. Example: after one round of pivoting in QuickSort, **two** of the items in the following array could be the pivot before:

55 12 **67** 99 76 **101** 111 102

So, how many of the following item could be the pivots?

45 67 88 66 97 101 98 102 111

(2 marks)

- 1
- 4
- 3
- 2
- 5

11. For a minheap, the extractMin operations needs ____ to restore the heap after deleting the root.

(2 marks)

- BubbleUp only

BubbleDown only
Tree Rotations
None of the answers here
Both BubbleUp and BubbleDown are possible

12. Converting a minheap to a maxheap in an array implementation, the best time complexity is

(2 marks)

O(log n)
O(n)
O(log log n)
O(n log n)
O(1) if we just treat the array as reversed

13. Given n integers and we want to find k smallest numbers. The fastest algorithm is

(2 marks)

Put the integers into an unsorted array. Find the min and marked it deleted k times
Sort the integers with mergeSort in ascending order and extract the first k numbers
Sort the integers with BubbleSort, but stop after k round before finishing
Put the integers into a minHeap and extractMin k times.
Put the integers into an AVL tree and extractMin k times

14. Which one of the following hash functions on integers will distribute keys most uniformly over 10 buckets numbered 0 to 9 for x ranging from 0 to 100000?

(2 marks)

h(x) = $(i^3)\%10$
h(x) = (11*i*i)%10
h(x) = (12*i)%10
h(x) = $(12*(i^3))\%10$
h(x) = $(i^2)\%10$

15. Fill in the blanks

(2 marks)

Given a hash table with open addressing and size m. If the number of item inserted is 0.8m, the expected number of probes will be __1__

Enter the correct answer below.

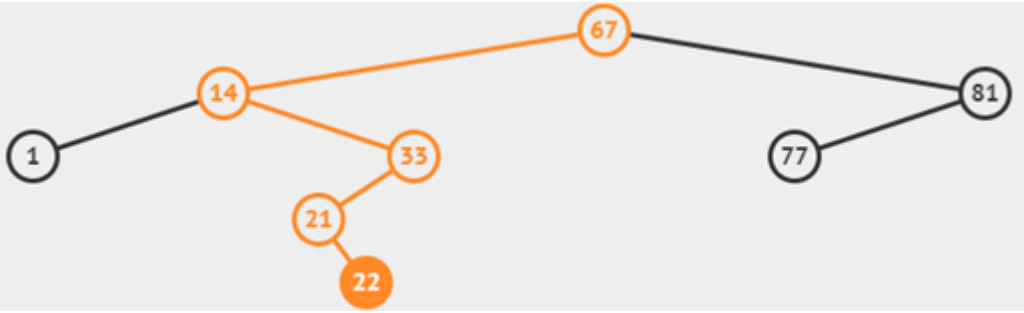
1 [ ] Please enter a number for this text box.

16. Suppose you have a linear probe hash table of size 11 with the hash function 'k%11'. Insert the keys: 41, 5, 40, 30, 19 into the table (in order). Which will be the hash table like?

(2 marks)

[19, NULL, NULL, NULL, NULL, 5, NULL, 40, 41, 30, NULL]
[NULL, NULL, NULL, NULL, 5, NULL, 40, 41, 30, NULL, 19]
[NULL, NULL, NULL, NULL, NULL, 5, NULL, 40, 41, 19, 30]
[30, NULL, NULL, NULL, NULL, 5, NULL, 40, 41, NULL, 19]
[NULL, NULL, NULL, NULL, NULL, 5, NULL, 40, 41, 30, 19]

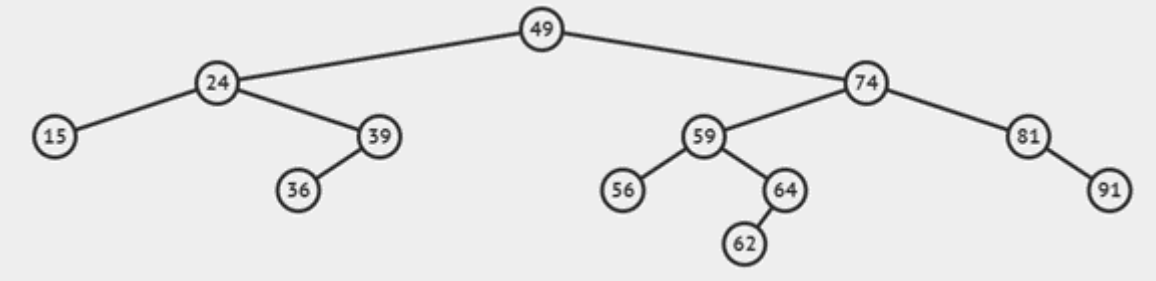
17. A new node 22 is added into the following AVL tree. How many rotations do we need to balance the tree?



(2 marks)

0
1
3
2
4

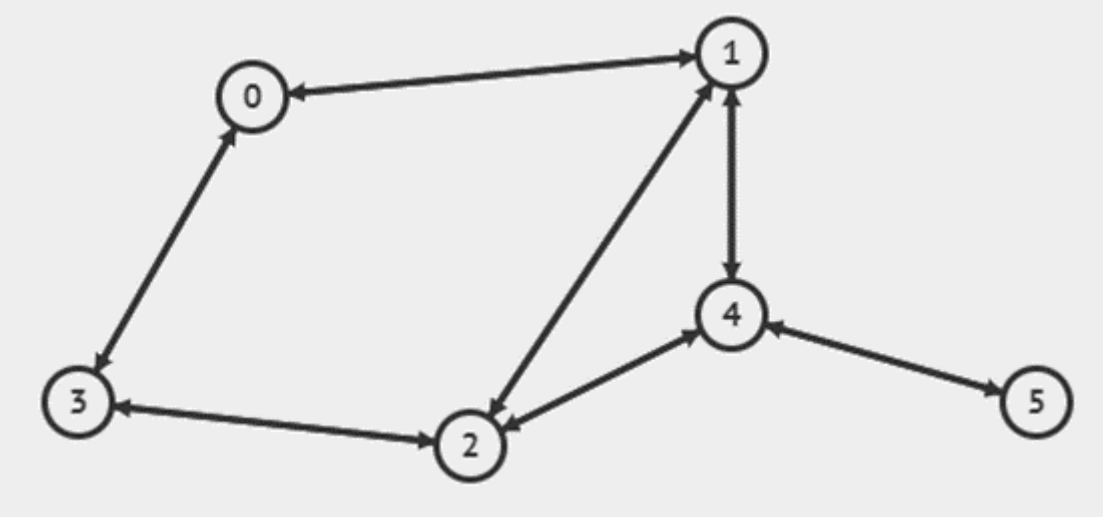
18. For the following AVL tree, how many rotations do we need to balance the tree after we deleted the node 15?



(2 marks)

1
3
4
2
0

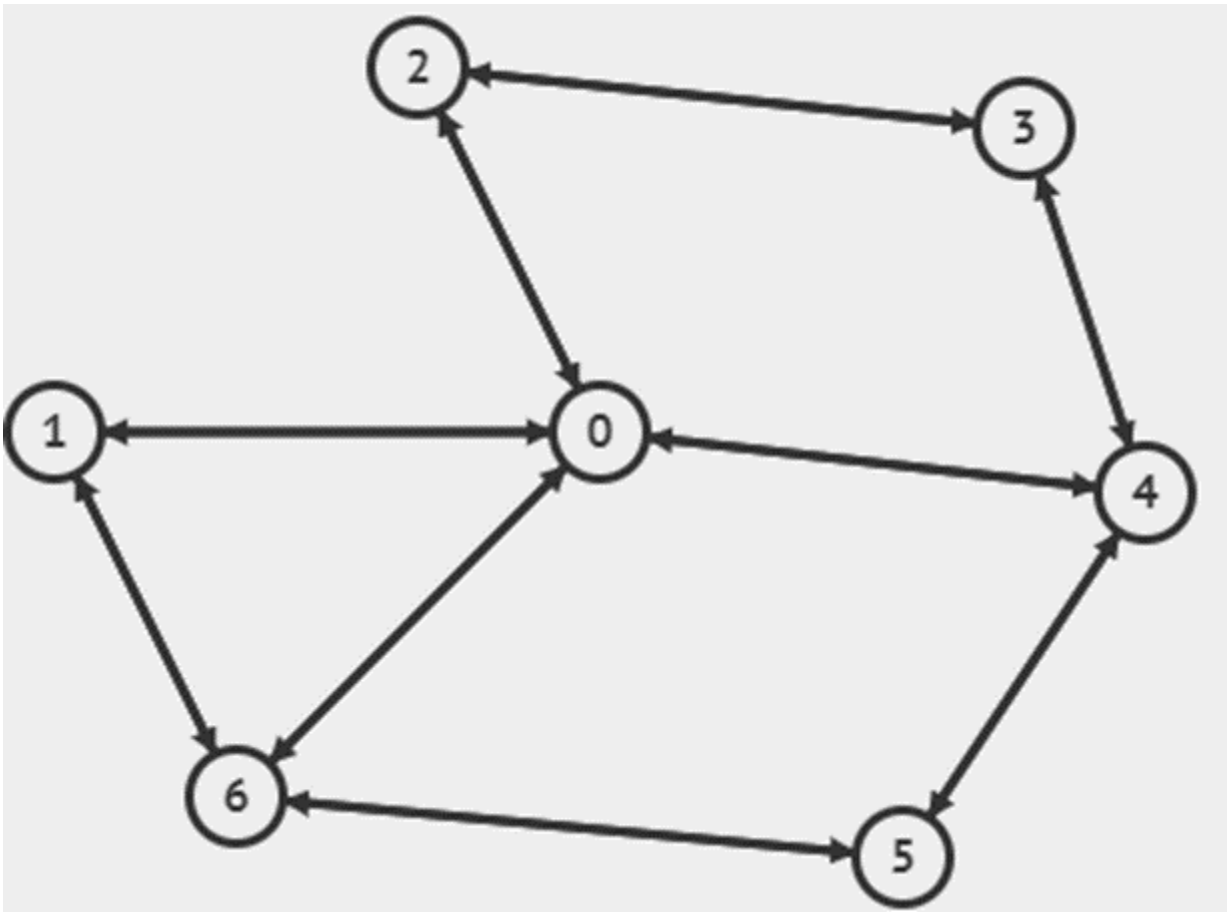
19. For the following graph, what is the possible BFS traversal order if we start from node 0?



(2 marks)

0,3,1,4,2,5
0,3,1,4,5,2
0,1,3,4,5,2
0,3,2,4,1,5
0,1,2,3,4,5

20. For the following graph, what is the impossible DFS traversal order if we start from node 0?



(2 marks)

0,6,1,5,4,3,2
0,4,5,6,1,3,2
0,4,3,2,5,6,1
0,1,6,5,3,2,1
0,6,5,1,4,3,2

21. Which of the following is true about Bellman-Ford SSSP algorithm?

(2 marks)

It is totally useless compared to other SSSP algorithm
The time complexity is the same as Dijkstra
It can be used to detect negative cycle in a graph
It can always finished in O(n) time with n is the number of vertices

It will never give the correct shortest path if a graph has some negative edges

22. Which of the following statement is true about the comparison between modified Bellman-Ford (BF) and Dijkstra? By "modified", it means BF will stop when there is no more improvement after one round of relaxation.

(2 marks)

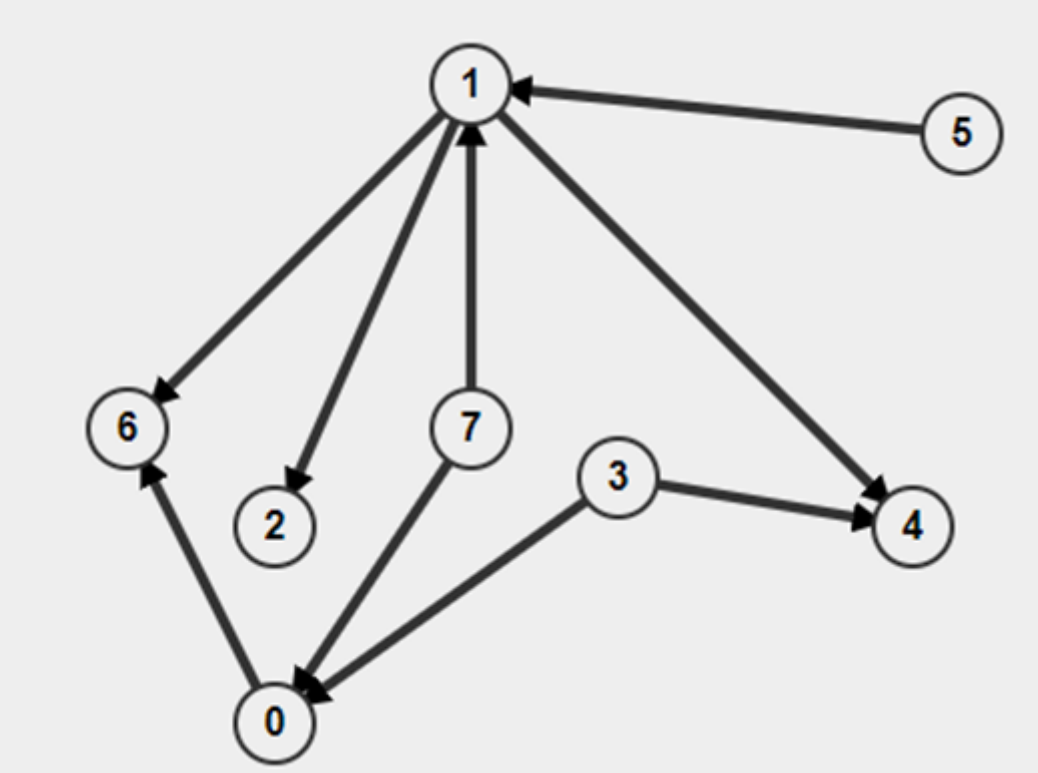
BF always runs faster than Dijkstra if the graph has some negative edges
BF can find the longest path in a graph and Dijkstra cannot
BF always runs faster than Dijkstra if the graph is a complete graph
BF always runs faster than Dijkstra if the graph is a star graph
BF cannot be used for directed graph but Dijkstra can.

23. Given a graph with n vertices and m edges, and all the weights of the edges are integers from 1 to 10. What will be the best time complexity for SSSP?

(2 marks)

O(m)
O(n log m)
O(m log n)
O(n)
O(m+n)

24. For the following graph, which one is NOT a possible topologicl sort ordering?



(2 marks)

5 7 1 2 3 0 6 4
7 5 1 3 4 0 6 2
5 7 3 1 0 2 6 4
5 7 3 0 1 4 6 2
3 7 0 5 2 1 4 6

25. Which of the following is NOT a property of an MST of a graph G?

(2 marks)

No cycles
For every cut of G, the minimum weight edge that crosses the cut is in the MST.
If you cut an MST, the two pieces are both MSTs of their subgraphs.
The shortest path between two node will be in MST always
For every cycle of G, the maximum weight edge is not in the MST

the MST.

26. Let G be a weighted graph with edge weights greater than 1 and G' be the graph constructed by squaring the weights of edges in G. Let T and T' be the minimum spanning trees of G and G', respectively, with total weights t and t'. Which of the following statements is TRUE?

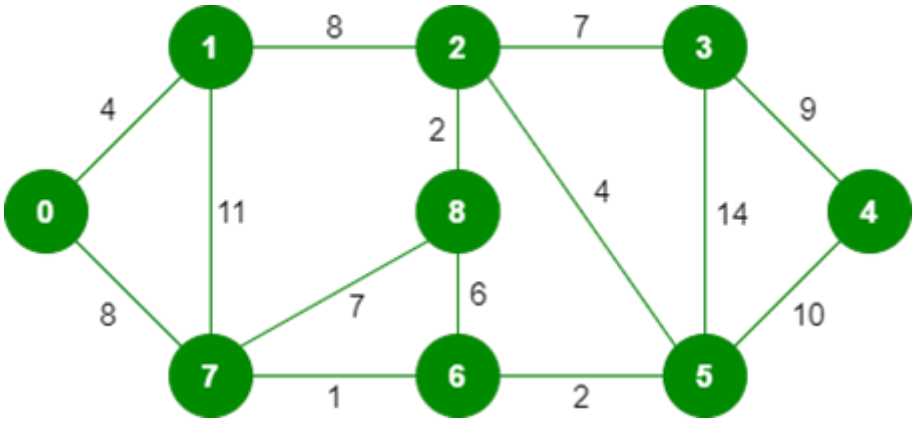
(2 marks)

- T'==t with total weight t'>t^2
- None of the answer here
- T'!=T but total weight t'=t^2
- T'==T with total weight t'<t^2
- T'==T with total weight t'=t^2

27. **Fill in the blanks**

(2 marks)

The weight of the MST of the following graph is 1 .



Enter the correct answer below.

1 Please enter a number for this text box.

28. Given a complete graph G with node 1 to N. The weight of each edge is the difference of the two node number (which is always a positive integer). The time to compute the MST of G is :

(2 marks)

- O(n log n)
- O(log n)
- O(n^2)
- O(n^2 log n)
- O(n)

29. For a country with a lot of citizens and they will change all the time. We have the data of age and salary for each individual. With an AVL tree and we want to do the following query: What is the average age for people with salary under a certain amount? We can use an AVL tree ordered by A and augmented each node by B. Provided every node already augmented with its number of node in its subtree.

(2 marks)

- A = ages, B = rank of the node
- A = ages, B = salary of each node
- A = ages, B = sum of salaries in the subtree
- A = salaries, B = sum of ages in subtree
- A = salaries, B = age of each node

30. Write down something you would let us (the teaching team) to know?

(2 marks)

Enter your answer here

Character Limit: 3000

Short Questions

31. What does the following function do if the input_stack is a stack of integers?

What is the time complexity of this function according to what?

```
Stack f(Stack input_stack) {
    Stack aux_stack
    while(!input_stack.empty())
    {
        int temp_value = input_stack.top()
        input_stack.pop()
        while(!aux_stack.empty() and aux_stack.top() > temp) {
            input_stack.push(aux_stack.top())
            aux_stack.pop()
        }
        aux_stack.push(temp_value)
    }
    return aux_stack
}
```

(4 marks)

Enter your answer here

Character Limit: 1000

32. If the key are integers and the hash table size is m = 1024. Is it a good hashing function? Give reason to support your answer. You may assume the input key k is uniformly distributed.

```
int hash(int k){
  return ((k*2)+47)%m;
}
```

(4 marks)

Enter your answer here

Character Limit: 1000

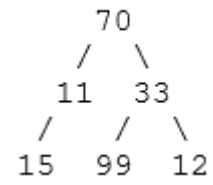
33. If you are given a binary tree and the tree node implementation is as follows:

```
template <class T>
class BinarySearchTree;

template <class T>
class TreeNode {
private:
        T _item;
        TreeNode<T>* _left;
        TreeNode<T>* _right;
        int _height;
        friend BinarySearchTree<T>;}
template <class T>
class BinarySearchTree {
private:
        int _size;
        TreeNode<T>* _root;
publich:
   void LOT();

}
```

And you can assume all attributes (left/right children, height, size, etc.) are set correctly. Write a function LOT() to print out the level order traversal of the tree with the input of the function as the root of the tree. E.g. if your tree is like this



Your output should be:

70 11 33 15 99 12

(4 marks)

Enter your answer here

Character Limit: 3000

34. Given a graph G and we want to compute the number of connected components of G. We can use either union-find or DFS/BFS. Compare their performances in terms of time and space complexities for their best and worst cases.

(4 marks)

Enter your answer here

Character Limit: 3000

35. You are given an ascending sorted array A of N unique numbers (int or float). Find any of the two consecutive numbers with their gap larger than the average gap size. Note that the average gap size is (A[N-1]-A[0])/(N-1). What is the time complexity of your algorithm?

(4 marks)

Enter your answer here

Character Limit: 3000

Long Questions (Dr Alan)

In this section, "DS" means "Data Structure".

36. You are the good wizard and your enemy trapped you in a magic jail. The magic jail consists of rooms and there are magic doors connecting the neighboring rooms. In order to escape from the jail, you need to visit all the rooms to break your enemy's spell.

However, each magic door needs you to use a certain amount of your magical energy, aka, mana. Your mana is only limited to an amount M. Can you decide which doors to be opened to break the spell? Or can you tell if your mana is not enough to escape from the jail?

- There are n rooms in the jail and there are D doors connected them.

- You can assume all rooms are connected by the doors

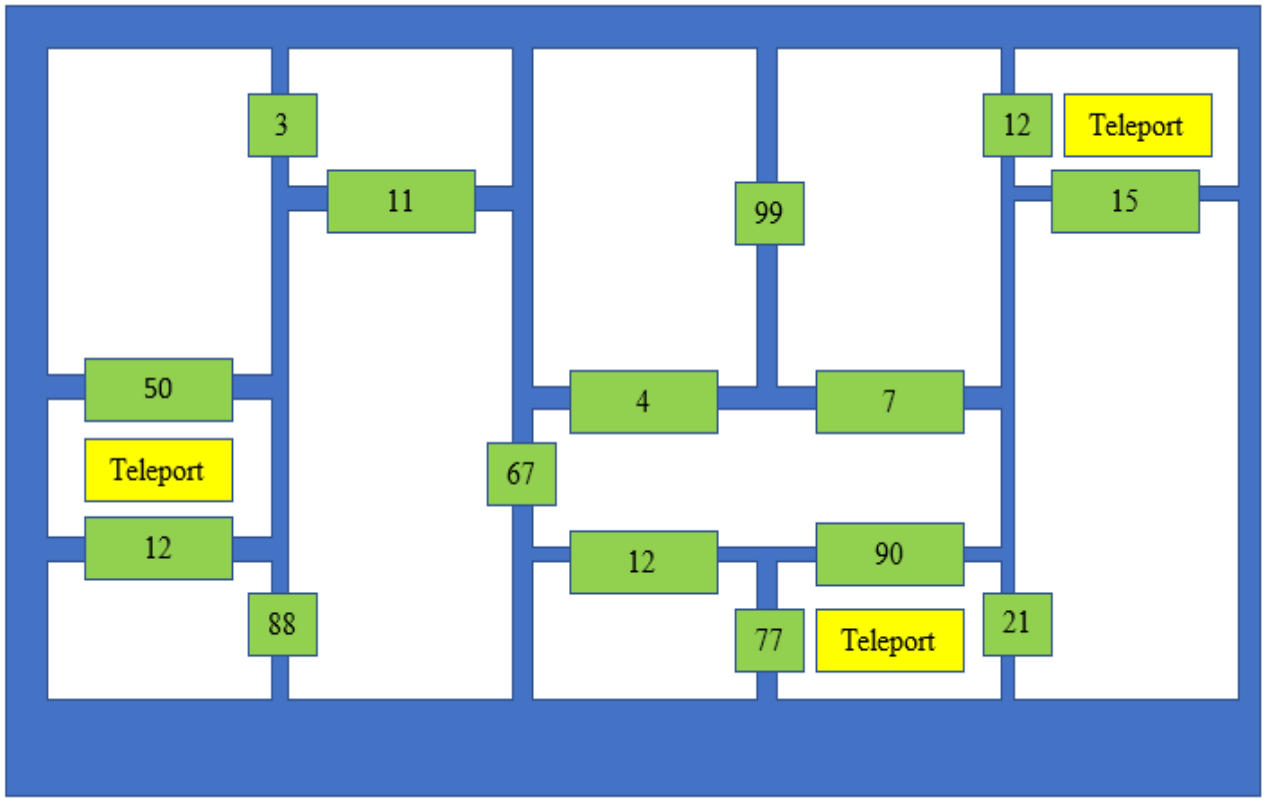
- Each door needs energy a[i] to open it, for i in 0..D-1

- You can assume you can start from any room you want

- There is a few rooms with a free teleport device that can let you to teleport to any other room that has a teleport device also without using any mana. And the number of teleport rooms is T.

- once any door is opened, it's opened forever.

Give the best algorithm to decide if your M unit of mana is enough for you to escape. What DS will you use to solve this problem and how will you model your DS? Also, what is the time complexity of your algorithm?



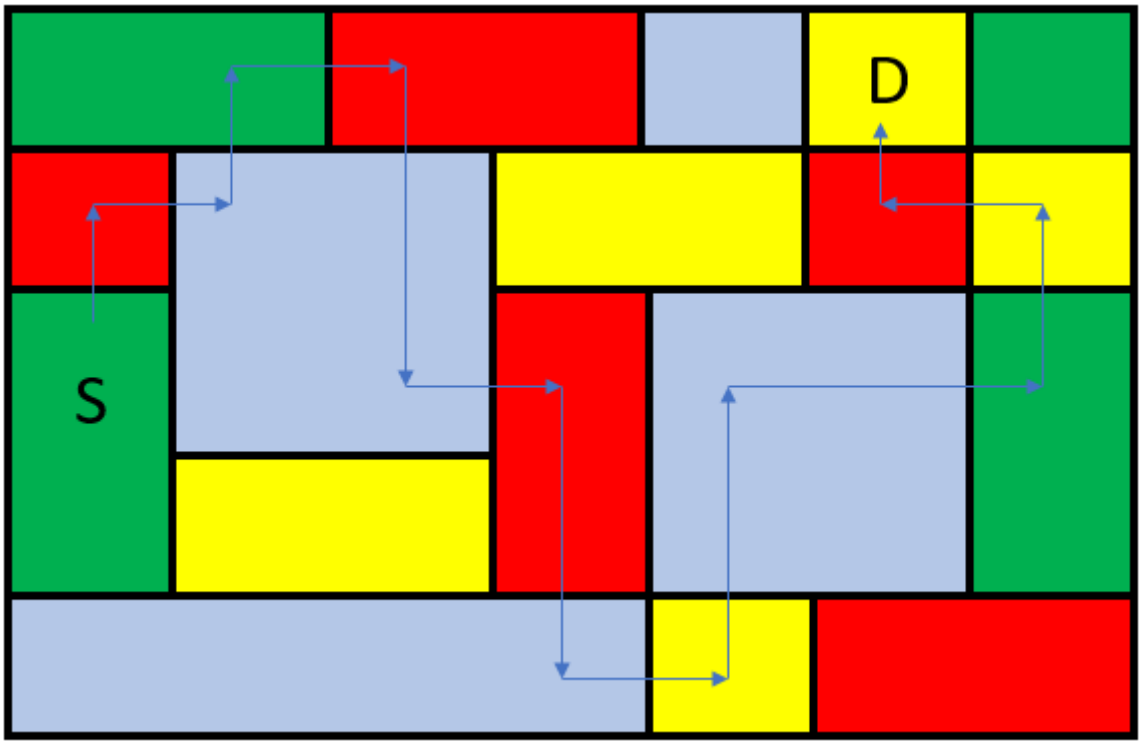
(6 marks)

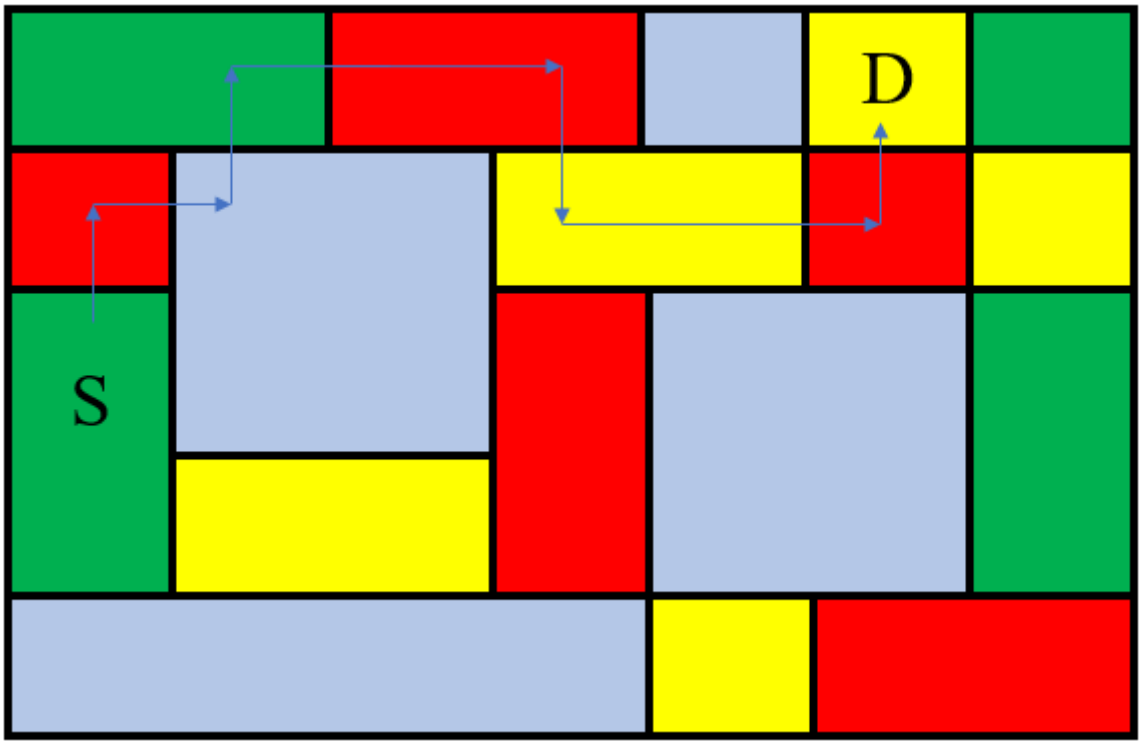
Enter your answer here

Character Word: 3000

37. Imagine you have a square board tiled with rectangular shapes (of varying heights and widths). Each has a cost associated with it. Find the cheapest path from S to D, hopping from rectangle to rectangle, subject to the following restriction: at each step of the path, you have to alternate a horizontal and a vertical move. You can assume that you can choose your direction at your starting point. By "cheapest path" we mean the fewest hops from room to room. The two diagrams below shows two paths, in which the lower diagram shows the cheaper path. Note that you may visit a room more than one time.

Give the best algorithm to solve this problem. What DS will you use and how will you model your DS?





(6 marks)

Enter your answer here

Character Limit: 3000

Long Question (Dr Steven)

If you reach this section (you cannot go back), it means that you are confident that you have scored most of the other 92 marks. Here is the last 8 marks.

38. In a mobile game Jetpack Joyride created by Halfbrick Studios in 2011, the main character Barry constantly runs across the screen (to the East) at the speed of **one pixel per second**. There are various dangerous obstacles along the way and Barry has to avoid them all.

In this simplified problem, let's assume that the game area is just 10 rows (row 0 at the bottom and row 9 at the top) but with **N** columns (column 0 on the left and column **N**-1 on the right) of pixels. **N** can be as big as 10^5 columns (it is a side scrolling game after all). A character 'X' denote dangerous obstacle and character '.' denotes passable pixel. Barry initially starts from row 0, column 0 (the bottom left side of the game area).

When the user presses the phone screen for one second, Barry activates his Jetpack and ascends to the neighboring North-East pixel until he reaches the top of the game area (row 9), when he will continue flying East until the user releases the screen.

When the user releases the phone screen, for each second, Barry glides down to the neighboring South-East pixel until he reaches the floor again (row 0), when he will again continue running to the East.

The objective of the game is to make Barry reach the last column **N**-1 safely (exactly after **N**-1 seconds as he runs, ascends, glides to E, NE, SE at speed of 1 horizontal pixel per second).

For example, given the following 10 rows and **N** = 11 columns game area:

```
.....XX...X
....XX...XX
...XX...XX.
..........
....XXX....
..........
.....X.....
....XX...X.
...XX...XX.
...X...XX..
```

Then Barry's path could have been as follows (highlighted as digits 0, 1, 2, ..., 9, A (to represent the 10th second)), run E, ascends to North-East direction for 4 seconds (to avoid the big obstacle), glides down to South-East direction for 2 seconds, ascends to North-East direction for another 2 seconds (to avoid the next obstacle), and finally glides down to South-East direction for the last second:

```
.....XX...X
....XX...XX
...XX...XX.
..........
....XXX....
.....5...9.
....4X6.8.A
...3XX.7.X.
..2XX...XX.
01.X...XX..
```

Note that there can be multiple valid paths that Barry can take, for example, this could have been Barry's valid move too:

```
.....XX...X
....XX...XX
...XX...XXA
.........9.
....XXX.8..
.....5.7...
....4X6....
...3XX...X.
..2XX...XX.
01.X...XX..
```

Now your task for this question is simple, given a 10 * **N** grid containing either 'X' (obstacle) or '.' (passable pixel), determine if Barry go from pixel (0, 0) - the bottom left pixel to pixel (**r**, **N**-1), i.e., any row **r** at the last column **N**-1?

a). If the game area is impossible to be completed by Barry, then simply output "IMPOSSIBLE".

b). Otherwise, print **any** valid path that Barry can take to complete the game.

Describe your proposed solution as pseudo-code and analyze its time complexity.

PS: An example game area with "IMPOSSIBLE".

```
.....XX...X
....XX...XX
...XX...XX.
..........
....XXX....
.....X.....
.....X.....
....XX...X.
...XX...XX.
...X...XX..
```

(8 marks)

Enter your answer here

Character Limit: 3000