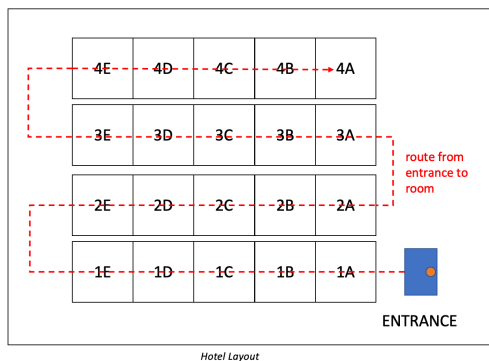


# Software Engineer

## QUESTION 1 (Software Design and Implementation)

An owner of a boutique hotel asked you to develop a program to help him assign available rooms to his guests. Your program must find the nearest available room measured by the route taken from the entrance to the room (see diagram) and assign it to the guest. Given that:

- The hotel has 4 floors and there are 5 rooms on each floor
- Rooms are numbered by the floor as prefix followed by a single alphabet suffix A, B, C, D or E where A is the first room (relative to the entrance) until E for the last room.



- There are 4 room statuses **Available**, **Occupied**, **Vacant** or **Repair** as follows:



1. Only **Available** rooms can be assigned to guests which will then become **Occupied**.



2. After the guest checks out of the room, the room becomes **Vacant**.



3. If housekeeping cleans a **Vacant** room, it becomes **Available**.



4. Housekeeping may take a **Vacant** room for repairs by marking the room as **Repair**.



5. Once a room is repaired, it will become **Vacant** again for cleaning.

6. Rooms under **Repair** can only be made **Vacant**.

7. **Available** and **Occupied** rooms cannot be repaired.

You can choose any method for input/output but must have:

- A method for requesting for room assignment, which reply with the assigned room number upon success.
- A method to check out of a room.
- A method to mark a room cleaned (Available).
- A method to mark a room for repair.
- A method to list all the available rooms.

This assignment is designed to test your familiarity with design principles, design patterns and ability to optimize for time complexity.

*Tips: There is at least 1 efficient algorithm that can find the nearest vacant room in  $O(1)$  time complexity.*

**Deliverables:**

1. Write a program with the language of your choice complete with unit tests
2. You will need to handle any edge cases or underspecified requirements reasonably.
3. Code will be judged on the thoughtfulness put into its design, the efficiency (time complexity) and OO design.
4. Instructions on how to compile, run and use your program.
5. Code must include frequent Git Commits.
6. Any other documentation if required.

## QUESTION 2 (Algorithms)

Given a large (more than 1,000,000) set of random numbers, implement an efficient data structure that can:

- Insert one number at a time.
- Search and return an element.
- Print all the number in ascending order.

The implemented data structure must satisfy the following:

- Insertion time complexity no worse than  $O(\log n)$
- Search time complexity no worse than  $O(\log n)$
- Print all number time complexity no worse than  $O(n)$

### **Deliverables:**

1. Write a program with the language of your choice complete with unit tests
  - a. The program returns an output in any format to represent the tree.
  - b. The program must also output the answer of the input string.
  - c. You are free to design your own models, inputs and outputs.
2. Code will be judged on its efficiency (time complexity) and readability.
3. Instructions on how to compile, run and use your program.
4. Code must include frequent Git Commits.
5. Any other documentation if required.

### **Important Notes**

- Submit your solutions to [hr@telepathy.ai](mailto:hr@telepathy.ai) with a link to a Github repo or a zip archive.
- If you are submitting a zip archive, please note that our email servers will block all emails with executables. So please make sure to remove all the binaries.
- Please email your answer directly. Do not share your answers with anyone else.
- Plagiarized submissions will be automatically disqualified.