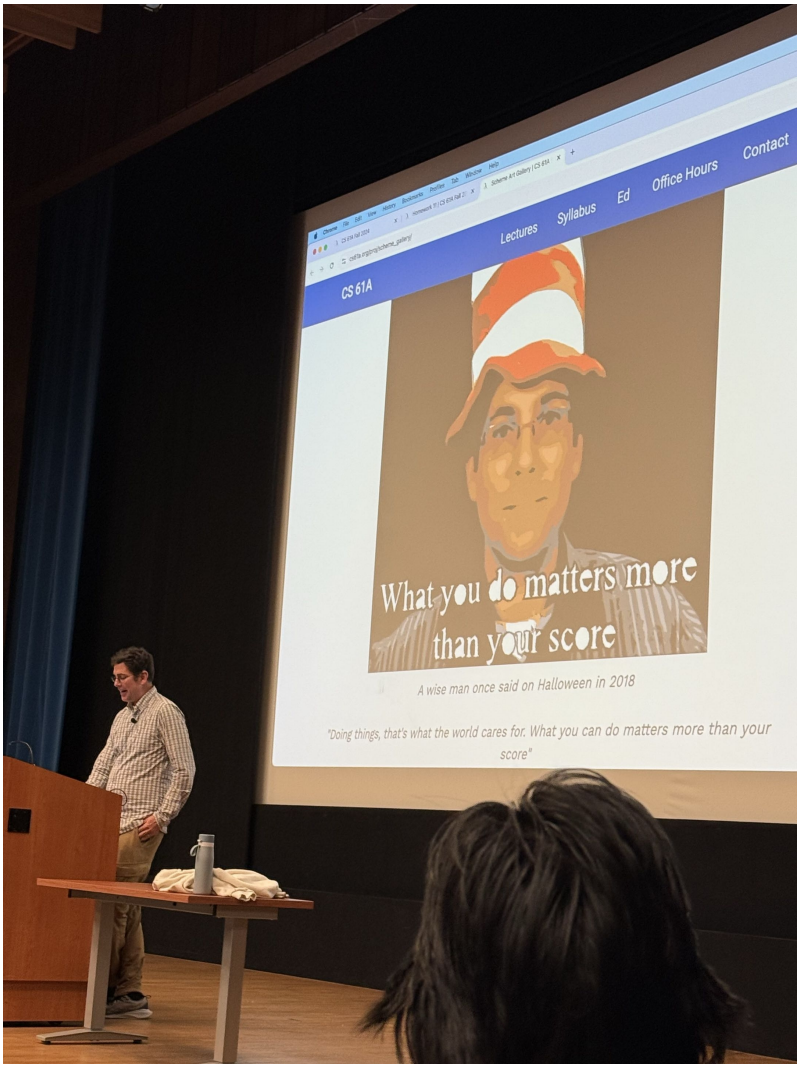
Problem 11: Average CS186 Class Experience
9+5 Points

Problem ID: `lecture`
Rank: `3+4`

Introduction

Big Ben and his friends got to class at berkeleyberkeleytime! Help them find the best places to sit so that they are close enough to hop on Clash Royale but also don't lose too much aura from the desks in Dwinelle.



Problem Statement

You are given an $\mathbf{N} \times \mathbf{M}$ grid representing the seat arrangement of a lecture hall. An occupied seat is marked with a hashtag `#`, and an available seat is marked with a dash `-`. You are also given an integer $\mathbf{K}$, representing a group of $\mathbf{K}$ people who need to find available seats.

Main Test Set

Your goal is to find $\mathbf{K}$ available seats such that the maximum Manhattan distance between any pair of people in the group is as small as possible. Formally, find distinct available coordinates $(r_1, c_1), (r_2, c_2), \cdots, (r_K, c_K)$ that solve the following expression:

$$\min(\max_{i \neq j} |r_i - r_j| + |c_i - c_j|)$$

You may output coordinates in any order, and if there are multiple optimal sets of coordinates, you may output any.

Bonus Test Set

For the bonus version, you must also optimize a secondary objective: minimizing the group's total aura loss. Among all the sets of $\mathbf{K}$ seats that successfully minimize the maximum Manhattan distance (the primary goal from the main version), you must choose the set that also minimizes the sum of the aura loss across all $\mathbf{K}$ selected seats.

The aura loss of a single seat is the number of occupied seats someone must squeeze past to reach it. Assume the only way to reach a seat is by entering its row from either the far left or the far right and walking straight towards it. Therefore, a seat's aura loss is simply the minimum of two values:

- The number of occupied seats strictly to its left in the same row
- The number of occupied seats strictly to its right in the same row

Output the coordinates of the $\mathbf{K}$ seats that solve the minimization problem. You may output the coordinates in any order, and if there are multiple optimal sets of coordinates, you may output any.

Input Format

The first line of the input contains a single integer $\mathbf{T}$ denoting the number of test cases that follow. For each test case:

- The first line contains three space-separated integers $\mathbf{N}$ $\mathbf{M}$ $\mathbf{K}$, denoting the dimensions of the seat grid and the number of people who need to sit down, respectively.
- The next $\mathbf{N}$ lines each contain a string of length $\mathbf{M}$ consisting of `#` and `-` characters, representing the rows of the seat grid.

Output Format

For each test case, output $\mathbf{K}$ lines representing the coordinates of the available seats that minimize the constraints above. Note that you may output these coordinates in any order.

Each line should consist of two space-separated integers r_i c_i, where r_i is the 0-indexed row number and c_i is the 0-indexed column number of the i^{th} seat.

Constraints

Time Limit: **2 seconds**
Memory Limit: **2048 MB**

$1 \leq \mathbf{T} \leq 10$
$1 \leq \mathbf{N}, \mathbf{M} \leq 10^3$
$1 \leq \mathbf{K} \leq \mathbf{N} \times \mathbf{M}$
$\mathbf{K}$ is less than or equal to the number of available seats.

The sum of $\mathbf{N} \times \mathbf{M}$ across all test cases does not exceed 10^6.

Sample Test Cases

Main Sample Input Download

```
3
3 3 2
#--
-##
##-
5 4 5
--##
#--#
####
#-##
#-##
2 7 2
#-#--##
#-#####
```

Main Sample Output Download

```
0 1
0 2

0 0
0 1
1 1
1 2
3 1

0 3
0 4
```

Note that this is one of many possible correct outputs. If there are multiple solutions, you may output any of them.

Main Sample Explanations

For test case #1, the two available seats in the first row enable the two friends to sit directly next to each other (a Manhattan distance of 1). This is the smallest possible distance between distinct seats, so we output the coordinates of these two seats.

For test case #2, the smallest possible maximum Manhattan distance for a set of 5 available seats is 4. For the coordinate set $\{(0,0),(0,1),(1,1),(1,2),(3,1)\}$, this is the distance between $(0,0)$ and $(3,1)$, which is $|0-3|+|0-1|=4$. Note that the coordinate set $\{(0,1),(1,1),(1,2),(3,1),(4,1)\}$ also achieves this Manhattan distance, so this would have been a valid solution as well.

For test case #3, there are two pairs of adjacent seats. Either pair is a valid solution for the main version of the problem.

Bonus Sample Input Download

```
3
3 3 2
#--
-##
##-
5 4 5
--##
#--#
####
#-##
#-##
2 7 2
#-#--##
#-#####
```

Bonus Sample Output Download

```
0 1
0 2
```

```
0 0
0 1
1 1
1 2
3 1
```

```
0 1
1 1
```

Note that this is one of many possible correct outputs. If there are multiple solutions, you may output any of them.

Bonus Sample Explanations

For test case #1, there is only one pair of seats that achieves the smallest maximum Manhattan distance of 1, so it is the only candidate for the aura loss minimization problem and thus the solution.

For test case #2, the smallest possible maximum Manhattan distance for a set of 5 available seats is 4. The coordinate sets $\{(0,0),(0,1),(1,1),(1,2),(3,1)\}$ and $\{(0,1),(1,1),(1,2),(3,1),(4,1)\}$ both achieve this Manhattan distance, so aura loss will be used as a tiebreaker. The aura losses of the first set are $\{0,0,1,1,1\}$ for a total of 3, while the aura losses of the second set are $\{0,1,1,1,1\}$ for a total of 4, so we must output the first set.

For test case #3, there are two pairs of seats that are directly next to each other. The set $\{(0,1),(1,1)\}$ has aura losses $\{1,1\}$ for a total of 2, while the set $\{(0,3),(0,4)\}$ has aura losses $\{2,2\}$ for a total of 4, so the first set is the only valid solution (as opposed to the main version).