

Description

Editorial

Solutions (462)

Submissions

2141. Maximum Running Time of N Computers

Hint

Hard

1.5K

40

☆

🔄

Companies

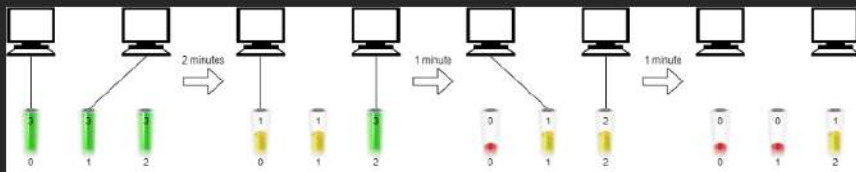
You have n computers. You are given the integer n and a **0-indexed** integer array `batteries` where the i^{th} battery can **run** a computer for `batteries[i]` minutes. You are interested in running **all** n computers **simultaneously** using the given batteries.

Initially, you can insert **at most one battery** into each computer. After that and at any integer time moment, you can remove a battery from a computer and insert another battery **any number of times**. The inserted battery can be a totally new battery or a battery from another computer. You may assume that the removing and inserting processes take no time.

Note that the batteries cannot be recharged.

Return the **maximum** number of minutes you can run all the n computers **simultaneously**.

Example 1:



Input: $n = 2$, `batteries = [3,3,3]`

Output: 4

Explanation:

Initially, insert battery 0 into the first computer and

i Java

• Auto

```
1 public class Solution {
2     public long maxRunTime(int n, int[] batteries) {
3         long sumPower = 0;
4         for (int power : batteries)
5             sumPower += power;
6         long left = 1, right = sumPower / n;
7
8         while (left < right){
9             long time = (left+right+1) / 2;
10            if (check(batteries, n, time))
11                left = time;
12            else
13                right = time - 1;
14        }
15        return left;
16    }
17
18    public boolean check(int [] B, int n, long time){
```

Testcase

Result

Accepted Runtime: 0 ms

• Case 1

• Case 2

Input

 $n =$

2

`batteries =``[3,3,3]`

Output

Console

🔍

Run

Submit

Description

Editorial

Solutions (10.1K)

Submissions

54. Spiral Matrix

Hint

Medium

12.8K

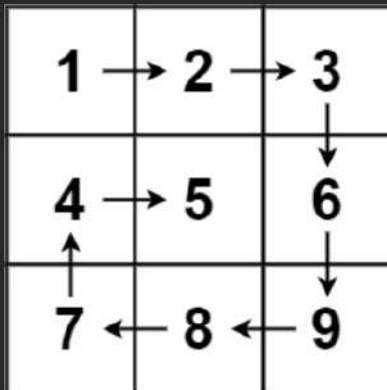
1.1K



Companies

Given an $m \times n$ matrix, return all elements of the matrix in spiral order.

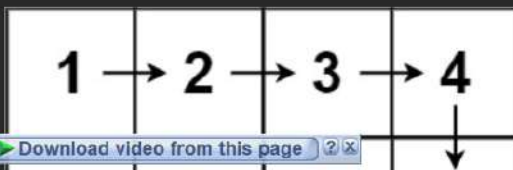
Example 1:



Input: matrix = [[1,2,3],[4,5,6],[7,8,9]]

Output: [1,2,3,6,9,8,7,4,5]

Example 2:

[Download video from this page](#)

Java

Auto

```
1 class Solution {
2     public List<Integer> spiralOrder(int[][] matrix) {
3         List<Integer> result = new ArrayList<>();
4         if (matrix == null || matrix.length == 0) {
5             return result;
6         }
7
8         int rows = matrix.length, cols = matrix[0].length;
9         int left = 0, right = cols-1, top = 0, bottom = rows-1;
10
11         while (left <= right && top <= bottom) {
12             for (int i = left; i <= right; i++) {
13                 result.add(matrix[top][i]);
14             }
15             top++;
16
17             for (int i = top; i <= bottom; i++) {
18                 result.add(matrix[i][right]);
```

Testcase

Result

Accepted Runtime: 0 ms

Case 1 Case 2

Input

matrix =

```
[[1,2,3],[4,5,6],[7,8,9]]
```

Output

```
[1,2,3,6,9,8,7,4,5]
```

Console



Run

Submit