Trilogy Questions (diff college)

1)

Worthy King II

Problem Description

You are the last successor of a worthy clan whose name we don't know (you can use your own imagination sometimes :p).

You are in your kingdom with no troops but your barracks train **X** troops every day. Initially **X = 1**.

There are **N** neighbor kingdom each of which has **A[i]** amount of troops guarding it. To take over a kingdom you need to attack it with at least **A[i]** troops.

Once you capture a kingdom, as a consequence of war, all of your troops die(even if you attacked with > A[i] troops) and **X** increases by **1**.

Assuming Attacking and capturing a kingdom takes no time, find minimum days the worthy king needs to wait to take over the entire globe?

Tip for Python users: Iteration is faster and more efficient than recursion.

Problem Constraints

1 <= N <= 20
1 <= A[i] <= 10^6

Input Format

First argument is an integer array A

Problem Constraints

1 <= N <= 20
1 <= A[i] <= 10^6

Input Format

First argument is an integer array A

Output Format

Return a single integer denoting minimum days

Example Input

Input 1:

```
A = [4, 3, 1]
```

Input 2:

```
A = [9, 9]
```

Example Output

Output 1:

```
4
```

2)

Horse Race

Flag Question

Problem Description

It's horse race day in the town and you have come to watch it with your friend. Your friend promises to give you money to bet on horses if you can solve his problem.

There are **N** horses in the race.

The town is weird, and so are the race rules. The finishing line for each horse might not be the same. The finish line for the **ith** horse is **A[i]** meters where **A** is an integer array. The horses start at 0. To finish the race, each horse must end at exactly the finishing line defined for that horse. If by chance, it moves further forward in a step, then it needs to come back to finish the race.

Also, the horses run in weird patterns. If a horse's stepping power is x, it can go in the positive direction y meters by spending 1 energy if y is odd and present in [1, x]. It can also go backward z meters by spending 1 energy if z is even and present in [1, x]. The stepping power of the **ith** horse is **B[i]** where **B** is an integer array.

But there is a catch, except for the very first movement of a horse, each positive direction movement **must** be immediately preceded by at least one negative direction movement.

Your friend is fascinated by horses that spend less energy. So, your friend asks, if he was to select one horse out of every **C consecutive** horses that spends the **least energy** out of the group to finish the race, what energy would the horses be spending?

Problem Constraints

|**A**| == |**B**| == N
1 <= **N** <= 10^5
5 <= **A[i]** <= 10^5
3 <= **B[i]** <= 10^5
1 <= **C** <= **N**

3)

Groovy's Walk

Flag Question

Problem Description

Groovy likes to have morning walks daily. He walks on an **A** x **A** matrix from start to finish. The start is at the top left cell of the matrix and the end is at the bottom right cell. He can only ever walk downwards or rightwards. That is, from the cell **(i, j)**, he can only go to **(i+1, j)** or **(i, j+1)**. He cannot go out of bounds of the array. **A** is guaranteed to be an odd integer.

But there is a problem, right at the centre of the matrix, there is a machine set up that produces a toxic gas that has spread to **B** outer rings apart from the center cell. That is to say, if **B** = 1 then all cells adjacent to the centre block, including which touch it on corners(the whole outer ring) are also toxic.

Your task is to determine the number of ways Groovy has, to reach the ending block from the starting block while avoiding the toxic cells. Since the number can be very large, output it modulo **1e9 + 7**.

For example, if **A** = 5 and **B** = 1, the grid will look like: (0 are passable cells while X are toxic cells)

```
00000
0XXX0
0XXX0
0XXX0
00000
```

Problem Constraints

A is guaranteed to be odd
3 <= A <= 1e5
0 <= B <= ⌊ A/2 ⌋ - 1 where ⌊ ⌋ stands for floor of value

4)

Count Unique Strings

Problem Description

You love strings a lot, so you decided to play the following game.
You have a tree **T** of **A** nodes. The tree is represented by a matrix **B** of dimensions **(A - 1) x 2**, such that there exist an edge between node **B[i][1]** and **B[i][2]**.
Each node is assigned a lowercase english character, which is represented by a string **C** of length **A.** Node **i** is assigned character at position **i** of string **C.**
You are given **Q** queries in the form of a matrix **D** of dimensions **Q x 2.** For each you query you will perform the following steps:

1. You will move from node **D[i][1]** to node **D[i][2]** using the shortest possible path.
2. Let **V[1], V[2] ... V[K]** be the nodes visited in the corresponding order. Create a string **S** such that length of **S** is equal to **K** and **S[i] = C[V[i]]**.
3. Store string **S** in your bag.

Return the number of unique strings you would create.

Problem Constraints

1 <= A <= 10^5
B is a matrix of dimensions (A - 1) x 2
1 <= B[i][1], B[i][2] <= A
length(C) = A
C only contains lowercase english alphabets
1 <= Q <= 10^5
D is a matrix of dimensions Q x 2
1 <= D[i][1], D[i][2] <= A

Output Format

Return a single integer.

Example Input

Input 1:

```
A = 5
B = [[1, 4], [5, 1], [2, 4], [3, 4]]
C = "baaba"
D = [[5, 2], [3, 5]]
```

Input 2:

```
A = 5
B = [[1, 2], [2, 3], [3, 4], [4, 5]]
C = "ababa"
D = [[1, 3], [3, 5], [2, 3], [2, 1]]
```

Example Output

Output 1:

```
1
```

Output 2:

```
2
```

Problem Constraints

$1 <= A <= 10^5$
B is a matrix of dimensions (A - 1) x 2
1 <= B[i][1], B[i][2] <= A
length(C) = A
C only contains lowercase english alphabets
$1 <= Q <= 10^5$
D is a matrix of dimensions Q x 2
1 <= D[i][1], D[i][2] <= A

Input Format

First argument A is an integer.
Second argument B is a matrix of integers.
Third argument C is a string
Fourth argument D is a matrix of integers.

Output Format

Return a single integer.