Homework - 5

IC100 Introduction To Programming

Question 1

You are given a 2D array of size n × n as input. Each value of the 2D array can either be 0 or 1. Suppose indices represent people and that the value at row i, column j of the array is 1 just in case i and j are friends and 0 otherwise.

(a) Define main() function such that it takes as input n, followed by n^2 entries of the 2D array. Per- form the tasks defined in part (b) and print the results in the format as shown in the examples in the main() function itself.

(b) Count how many pairs of friends are represented in the 2D array.

```
5
0 1 0 1 1
1 0 1 0 1
0 1 0 0 0
1 0 0 0 1
1 1 0 1 0
```

Output:

```
Number of pairs of friends = 6
```

Question 2

You are given an array of integers of size n. Write a function check () that returns true, if and only if the number of occurrences of each value in the array is unique. For all other cases, it must return false.
Note: Print the result in the main() function.
Example:
Input
10
1 2 2 3 3 3 4 4 4 4
Output

true
Explanation: The value 1 has one occurrence, 2 has two occurrences, 3 has three occurrences and 4 has four. No two values have the same number of occurrences in the array.

Question 3

Suppose you have an array cost[ ] of size n where cost[i] is the cost of the i^{th} step on a staircase. Once you pay the cost, you can either climb one or two steps of the stairs. You can either start from the step with index 0 or step with index 1. Write a function minCost(int cost[ ], int n) that computes and returns an integer value which is the minimum cost that is incurred in reaching the top floor.
Note: Print the result in the main() function.
Example
Input:
3
10 15 20
Output:
Minimum Cost = 15
Explanation: Cheapest is: start on cost[1], pay that cost, and go to the top.
Input:
10
1 100 1 1 1 100 1 1 100 1
Output:
Minimum Cost = 6
Explanation: Cheapest is: start on cost[0], and only step on 1s, skipping cost[3].
Hint: minCost[i] = cost[i] + min(minCost[i - 1], minCost[i - 2]) for i >= 2.
minCost[0] = cost[0], minCost[1] = cost[1]

Question 4

Watson gives Sherlock an integer array A of size n. His challenge is to find if there is an element in the array such that the sum of all elements to its left is equal to the sum of all elements to its right. Write a function findElement(int A[ ], n) that takes Array A and integer n as input and return 1 if such element exists else return 0. Example:
Input :
3 1 2 3
Output :
0
Explanation : No such element is there.

Question 5

Let's call an array A of size n, a mountain if the following properties hold:
n >= 3
There exists some i with 0 < i < n - 1 such that:
A[0] < A[1] < ... < A[i - 1] < A[i]
A[i] > A[i + 1] > ... > A[n - 1]
You are given an array A of size n that is guaranteed to be a mountain. Write a function peakIndex(int A[ ], int n) that takes as parameters array A and integer n and returns any i such that A[0] < A[1] < ... < A[i - 1] < A[i] > A[i + 1] > ... > A[n - 1]. Use ideas involved in Binary Search to solve this problem.

Submission

Please submit your homework in piazza under hw5 folder and make it a private submission to the instructors. Zip all the codes and name the zip as yourname_rollno
Submission deadline is 8:00pm Jan 16.