

CST 370 – Homework 6

Due: 2:00pm on May 7, 2019

Final Submission Deadline: 2:00pm on May 14, 2019

Changelog:

- 4/29/19: Removed unnecessary text from dynamic programming input sections.

About

This homework assignment asks you to solve various problems using either Prim's or Kruskal's algorithms or Dynamic Programming Techniques.

There are 6 problems in this homework **but you don't need to do all of them!** You must complete 40 points worth of problems. You will get credit for solving a dynamic programming problem (*either iteratively or recursively*). If you solve a dynamic programming problem *both iteratively and recursively*, you will get extra credit. (It'll count as an extra credit problem: 0.5% of final grade).

There are different hackerrank challenges for the iterative version and recursive version of each dynamic programming problem. Please use the appropriate challenge when submitting your work.

Submissions

<https://www.hackerrank.com/contests/cst370-s19-hw6/challenges>

The problems for this homework will be hosted on HackerRank where automated tests will run. You may submit code on HackerRank as many times as you'd like. Once you are comfortable with your code, you can copy the submission link for that problem and include it in the iLearn assignment for me to grade. **I will only grade hackerrank links submitted on iLearn.** iLearn has videos describing how to get a submission link.

After the due date, you may continue working on problems on HackerRank. If you submitted something at the initial deadline, you may submit HackerRank links again on the iLearn assignment for resubmissions. Resubmissions will be assessed a 10% deduction.

Scoring

Your score on HackerRank or Kattis will give you *an approximation* of what your grade will be; however, I will read your code and determine your final grade myself, as detailed on the syllabus.

Problems

- | | | |
|---|---------------------|-----------|
| 1 | Earthquake Recovery | 10 points |
| 2 | Coin change | 10 points |
| 3 | Leap Frog | 10 points |
| 4 | Maximum Sub-Square | 10 points |
| 5 | Climbing Stairs | 10 points |
| 6 | Meeting Scheduling | 10 points |

1. Earthquake Recovery (10 points)

<https://www.hackerrank.com/contests/cst370-s19-hw6/challenges/hw6-earthquake>

An Earthquake has knocked out a lot of roads in a county so citizens are cut off from essential resources. The county's repair crews are working overtime to get everyone reconnected to the road network. You are given the towns and roads in the county, as well as the amount of time it will take to repair damaged roads. You will find the minimum amount of time it will take to get everyone reconnected?

Input

- The first line will contain an integer n , the number of towns that are stranded.
- The next n lines will contain those towns.
- The next line will contain an integer r , the number of roads in need of repair.
- The next r lines will contain the roads in the format of comma-separated endpoints, followed by an integer, the amount of time required to repair it.

All towns must be reachable after repairing the roads. Any town that is not unconnected is a valid reconnection point. This input is very similar to the input for homework 5's Essential Services problem. You can reuse your input handling code that problem on this problem.

Constraints

You can assume all the weights are positive (it takes time to repair a road) and that all towns have unique names.

Output

A single line containing an integer, the amount of time it will take to get everyone reconnected.

Sample Input 1

```
6
Pacific Grove
Salinas
Monterey
Seaside
Marina
CSUMB
9
Pacific Grove, Monterey, 80
Pacific Grove, Pebble Beach, 10
Pacific Grove, CSUMB, 20
Monterey, Marina, 15
Monterey, Salinas, 30
Marina, Salinas, 45
Seaside, CSUMB, 25
Seaside, Marina, 15
CSUMB, Marina, 10
```

Sample Output 1

```
100
```

2. Coin change (10 points)

You will be given the denominations of coins in a monetary system and a required sum. Your job is to find the number of ways you can make change for that sum assuming order of coins does not matter.

Submission

- Use this link to submit a **iterative** dynamic programming solution to this problem:
<https://www.hackerrank.com/contests/cst370-s19-hw6/challenges/hw6-coinchange-iterative>
- Use this link to submit a **recursive** dynamic programming solution to this problem:
<https://www.hackerrank.com/contests/cst370-s19-hw6/challenges/hw6-coinchange-recursive>

Input

- The first line will contain an integer, n , the required sum.
- The second line will contain an integer, c , the number of coins in the monetary system.
- The third and final line will contain c space-separated integers, the denominations of those coins.

Constraints

For simplicity, you can assume the required sum and all coin denominations are integers.

Output

The output will consist of a single line containing an integer, the number of ways to make the required sum.

Sample Input 1

```
25
4
10 5 25 1
```

Sample Output 1

```
13
```

Sample Input 2

```
100
4
10 5 25 1
```

Sample Output 2

```
242
```

This page is intentionally left blank.

3. Leap Frog (10 points)

You will be given an array of integers representing the lily pads. Each integer indicates how many lily pads forward the frog can jump in a single leap from that lily pad. Find the minimum number of jumps the frog needs to reach shore (past the lily pads, i.e., the end of the array).

Submission

- Use this link to submit a **iterative** dynamic programming solution to this problem: <https://www.hackerrank.com/contests/cst370-s19-hw6/challenges/hw6-leapfrog-iterative>
- Use this link to submit a **recursive** dynamic programming solution to this problem: <https://www.hackerrank.com/contests/cst370-s19-hw6/challenges/hw6-leapfrog-recursive>

Input

- The first line will contain a single integer n , the number of lily pads.
- The second line contains n space-separated integers, the lily pads in order across the pond.

Constraints

You can assume the frog starts on lily pad 0 (the first in the array). You can also assume there are no negative numbers in the array.

Output

A single line consisting of either

- a single integer, the minimum number of jumps.
- -1 if the Frog can't reach shore and should turn back.

Sample Input 1

```
11
1 3 5 8 9 2 6 7 6 8 9
```

Sample Output 1

```
3
```

Sample Input 2

```
6
1 0 10 5 11 3
```

Sample Output 2

```
-1
```

This page is intentionally left blank.

4. Maximum Sub-Square (10 points)

You will receive a square binary matrix representing unoccupied and occupied space in a park.

You are trying to install a square garden and would like it to be as large as possible. Find the dimension of the largest possible garden (i.e., the largest sub-square of just 0s, vacant space).

Submission

- Use this link to submit a **iterative** dynamic programming solution to this problem:
<https://www.hackerrank.com/contests/cst370-s19-hw6/challenges/hw6-subsquares-iterative>
- Use this link to submit a **recursive** dynamic programming solution to this problem:
<https://www.hackerrank.com/contests/cst370-s19-hw6/challenges/hw6-subsquares-recursive>

Input

- The first line will contain an integer, n , the dimension of the park.
- The next n lines will contain n space-separated bits (0 or 1), indicating whether that part of the park is vacant or occupied.

Constraints

You can assume n is positive.

Output

A single line containing an integer, the maximum dimension for the garden.

Sample Input 1

```
4
0 1 0 0
1 0 0 0
0 0 0 0
0 0 1 0
```

Sample Output 1

```
2
```

Sample Input 2

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

Sample Output 2

```
1
```

This page is intentionally left blank.

5. Climbing Stairs (10 points)

You have to climb a special staircase. It costs money to stand on each step so you want to **minimize the cost** incurred by climbing the stairs. You are given an array of integers representing the cost to stand on the i^{th} step where i is the index. **You may start on either step 0 or 1 and you can go up 1 or 2 steps each time.** **You must end on the last step.**

Submission

- Use this link to submit a **iterative** dynamic programming solution to this problem: <https://www.hackerrank.com/contests/cst370-s19-hw6/challenges/hw6-stairs-iterative>
- Use this link to submit a **recursive** dynamic programming solution to this problem: <https://www.hackerrank.com/contests/cst370-s19-hw6/challenges/hw6-stairs-recursive>

Input

- The first line will contain an integer n , the number of steps.
- The second and final line will contain n space-separated integers, the cost of stepping on each step.

Constraints

You can assume the cost of a step is non-negative.

Output

The output will consist of a single integer, n , the cheapest cost you can climb the stairs for.

Sample Input 1

```
5
7 10 15 3 11
```

Sample Output 1

```
24
```

Sample Input 2

```
12
11 9 2 4 15 19 3 25 18 8 0 1
```

Sample Output 2

```
48
```

This page is intentionally left blank.

6. Meeting Scheduling (10 points)

My company is perpetually short on meeting rooms. A number of meetings have been scheduled for the same room but unfortunately they overlap in time so not all of them can occur. Each meeting has a start and end time and a priority (importance). Find the maximum cumulative priority we can accomplish with non-overlapping meetings.

Submission

- Use this link to submit a **iterative** dynamic programming solution to this problem: <https://www.hackerrank.com/contests/cst370-s19-hw6/challenges/hw6-meetings-iterative>
- Use this link to submit a **recursive** dynamic programming solution to this problem: <https://www.hackerrank.com/contests/cst370-s19-hw6/challenges/hw6-meetings-recursive>

Input

- The first line of the input will be an integer, n , the number of meetings.
- The next n lines will consist of three space separated integers: the start time, the end time, and the priority, in that order.

Constraints

You can assume all the priorities are positive.

Output

The output will consist of a single line containing an integer, the maximum cumulative priority we can get scheduled.

Sample Input 1

```
7
8 9 5
11 13 4
10 13 2
11 15 6
15 16 8
13 14 3
8 11 7
```

Sample Output 1

```
22
```

This page is intentionally left blank.