[CS 11 25.1] Lab 6b – Amelia the Adventurer

Cheatsheet is available here: https://oj.dcs.upd.edu.ph/cs11cheatsheet/

Submit solution [CS 11 25.1] Lab Exercise 6

My submissions

Points: 200 (partial)
Time limit: 12.0s
Memory limit: 1G

Problem type

Allowed languages
py3

Problem Statement

Amelia is circumnavigating the earth on her plane.

The toughest leg of this journey involves going across the Pacific ocean. Luckily, there are n islands along the way, conveniently situated along a straight line, which we can model as the real number line. Each of the n islands is located along this line, and also has a **risk value** measuring how risky it is to land on that island.

We number the islands 0 to $n-1$. Amelia starts at the island numbered s and needs to reach the island numbered e. She doesn't have to land on every island. However, her plane can only travel up to a distance of d units between islands, which means she may have to land on some islands along the way to refuel. Other than that, she is free to choose which islands to land on, and in what order.

We define the **overall risk** of her trip as the maximum risk value among the islands she landed on, including the starting island s and the destination island e.

If she plans her trip optimally, what is the minimum risk in which Amelia can go from island s to island e?

Task Details

Your task is to implement a function called `flight_safety`. The function takes four arguments:

- the first is a `tuple` or `list` of n pairs of `int`s describing the islands. In each pair,
  - the first element is the location of the island on the real number line.
  - the second element is the risk value for that island.
- the second is the `int` s.
- the third is the `int` e.
- the fourth is the `int` d.

The function must return either an `int` denoting the minimum overall risk, or `None` if it is impossible to reach island e.

Restrictions

Note that some names are banned. Here are a few of them: `input`, `type`. This is not an exhaustive list. (If you accidentally use a variable name that turns out to be banned, please rename it.)

The following imports are now allowed:

- `count`, `islice`, and `chain` from `itertools`.

For this problem:

- The source code limit is 2000.

Example Calls

Example 1 Function Call

```
flight_safety([
    (19, 1),
    (9, 4),
    (5, 2),
    (14, 5),
    (11, 1),
    (17, 3),
    (3, 2),
    (1, 1),
], 5, 2, 6)
```

Copy

Example 1 Return Value

```
3
```

Copy

Example 2 Function Call

```
flight_safety([
    (19, 1),
    (9, 4),
    (5, 2),
    (14, 5),
    (11, 1),
    (17, 3),
    (3, 2),
    (1, 1),
], 5, 2, 5)
```

Copy

Example 2 Return Value

```
5
```

Copy

Example 3 Function Call

```
flight_safety([
    (19, 1),
    (9, 4),
    (5, 2),
    (14, 5),
    (11, 1),
    (17, 3),
    (3, 2),
    (1, 1),
], 5, 2, 3)
```

Copy

Example 3 Return Value

```
None
```

Copy

Constraints

- The function `flight_safety` will be called at most $60{,}000$ times.
- The sum of the ns will be at most $250{,}000$.
- $1 \le n \le 250{,}000$
- $0 \le s, e < n$
- Each location is a nonnegative integer at most 10^{20}.
- Each risk value is a nonnegative integer at most 10^{20}.
- $0 \le d \le 10^{20}$

Scoring

Note: New tests may be added and all submissions may be rejudged at a later time. (All future tests will satisfy the constraints.)

- You get 60 🔴 points if you solve all test cases where:
  - $n \le 8$
  - The sum of the ns is at most 40.
  - it is always possible to go from island s to island e.
- You get 40 🔴 points if you solve all test cases where:
  - $n \le 8$
  - The sum of the ns is at most 40.
- You get 30 🔴 points if you solve all test cases where:
  - $n \le 50$
  - The sum of the ns is at most 500.
- You get 20 🔴 points if you solve all test cases where:
  - $n \le 4{,}000$
  - The sum of the ns is at most $8{,}000$.
- You get 50 🔴 points if you solve all test cases.

For the first category above, you only get 10 points if you solve the smoke tests but not all the tests in that category.

Clarifications

Report an issue

No clarifications have been made at this time.