Problem Set 3 Exercise #16: Rabbit Jumps [Hard]

Reference: Lecture 6 notes

Learning objectives: One-dimensional array; Algorithm design

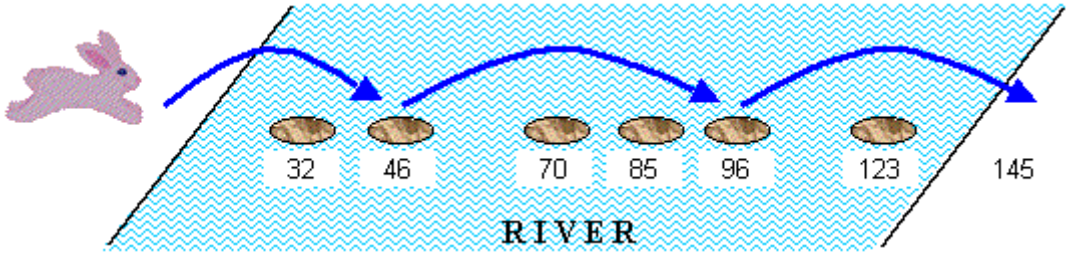
Estimated completion time: 70 minutes

Problem statement:

[CS1101X AY2005/06 Semester 1 PE, Q1]

A bunny can hop at most 50 centimetres far. It wants to cross to the other side of the river, but it cannot swim. So the only hope is to hop on the rocks on the river, which are positioned in a straight line. The positions of the rocks are measured from the start location, assuming that the bunny starts at location zero. The opposite bank could be treated as a big rock.

In the figure below, the rocks are at locations 32, 46, 70, 85, 96, 123, and the opposite riverbank at location 145.



The bunny will jump as far as it could for each hop. **What is the least number of jumps it needs to take to reach the other side of the river?** For the above example, it needs to make 3 jumps, as shown in the figure above.

Write a program **PS3_Ex16_RabbitJump.java** that reads in a positive integer *n* that represents the number of rocks (including the opposite riverbank), and then on the next line *n* distinct positive integers in ascending order that represent the positions of the rocks. Your program should output the minimum number of jumps needed, or "Impossible" if it is not possible for the rabbit to reach the other side of the river.

Your program should include a static method

```
int countJumps(int[] rocks)
```

to compute and return the number of jumps required, or -1 if it is not possible for the rabbit to reach the other side of the river.

Sample run #1:

```
Enter the number of rocks: 7
Enter locations of 7 rocks: 32 46 70 85 96
123 145
3 jumps
```

Sample run #2:

```
Enter the number of rocks: 5
Enter locations of 5 rocks: 40 70 150 160 180
Impossible
```

Sample run #3:

```
Enter the number of rocks: 11
Enter locations of 11 rocks: 30 70 75 120 160
170 180 190 200 246 258
7 jumps
```