

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 import java.util.Arrays;
4
5 class Program {
6     // O(n^2) time | O(n) space
7     public static int minNumberOfJumps(int[] array) {
8         int[] jumps = new int[array.length];
9         Arrays.fill(jumps, Integer.MAX_VALUE);
10        jumps[0] = 0;
11        for (int i = 1; i < array.length; i++) {
12            for (int j = 0; j < i; j++) {
13                if (array[j] >= i - j) {
14                    jumps[i] = Math.min(jumps[j] + 1, jumps[i]);
15                }
16            }
17        }
18        return jumps[jumps.length - 1];
19    }
20 }
21
```

Solution 1Solution 2Solution 3

```
1 class Program {
2     public static int minNumberOfJumps(int[] array) {
3         // Write your code here.
4         return -1;
5     }
6 }
7
```

Our Tests

Custom Output

Submit Code

```
1 class Program {
2     // ...
3     public static int minNumberOfJumps(int[] array) {
4         // ...
5     }
6 }
```

```
1 // Custom Output
2
```

```

10 #Print
11 #Print testPasses()
12 test() test = 10, 50
13 #Print testPasses() and test() test = 10
14 }
15
16 #Print
17 #Print testPasses()
18 test() test = 10, 50
19 #Print testPasses() and test() test = 10
20 }
21
22 #Print
23 #Print testPasses()
24 test() test = 10, 5, 50

```

Run or submit code when you're ready.