

2. Parking Dilemma

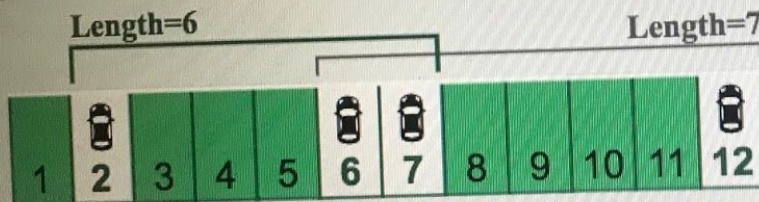
There are many cars parked in a parking lot. The parking lot is a straight line with a parking spot for every meter. There are n cars currently parked and a roofer wants to cover them with a roof. The requirement is that at least k cars are covered by the roof. Determine the minimum length of the roof that will cover k cars.

Example

$n = 4$

$\text{cars} = [6, 2, 12, 7]$

$k = 3$



Two roofs that cover three cars are possible: one covering spots 2 through 7 with a length of 6, and another covering slots 6 through 12 with a length of 7. The shortest roof that meets the requirement is of length 6.

Function Description

Complete the function `carParkingRoof` in the editor below.

`carParkingRoof` has the following parameter(s):

`int cars[n]`: the parking spots where cars are parked

`int k`: the number of cars that have to be covered by the roof

Returns:

`int`: the minimum length of a roof that can cover k cars

```
17 #
18
19 def carParkingRoof(cars, k):
20     cars.sort()
21     minLength = cars[len(cars)-1] - cars[0] + 1 #this is the
max possible length
22     for i in range(len(cars) - k):
23         currLength = cars[i+k-1] - cars[i] + 1
24         print(cars[i+k-1])
25         print(cars[i])
26         print(currLength+1000)
27         if currLength < minLength:
28             minLength = currLength
29     return minLength
30
```