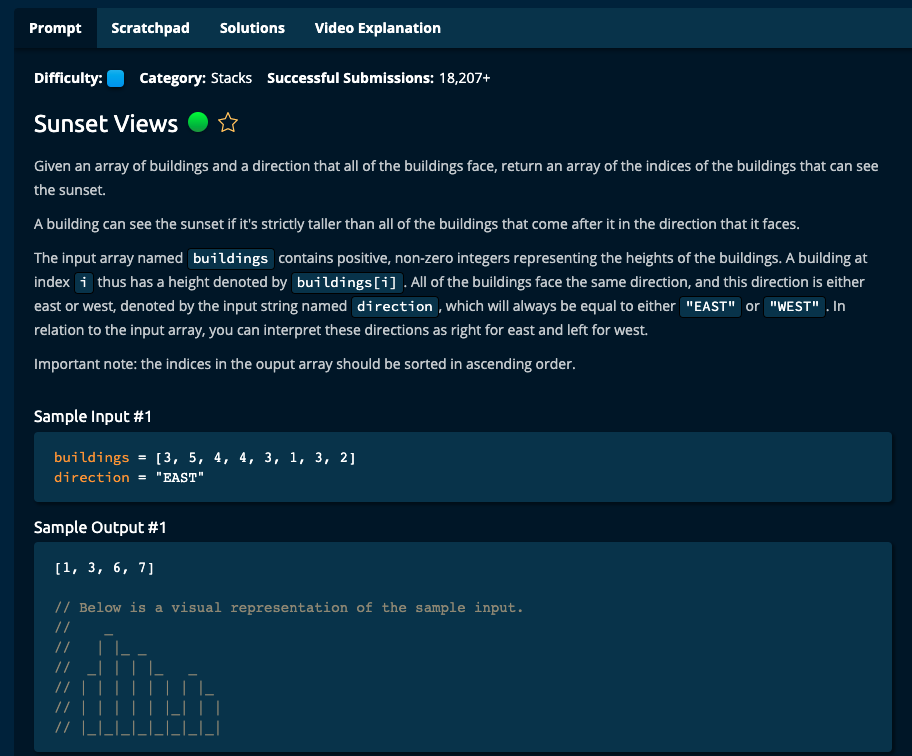
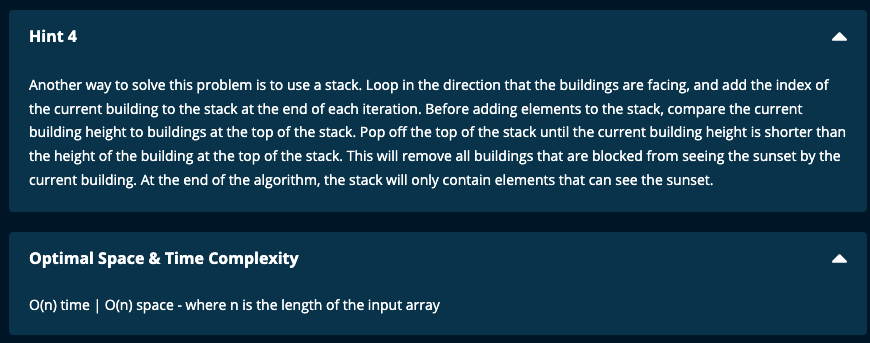
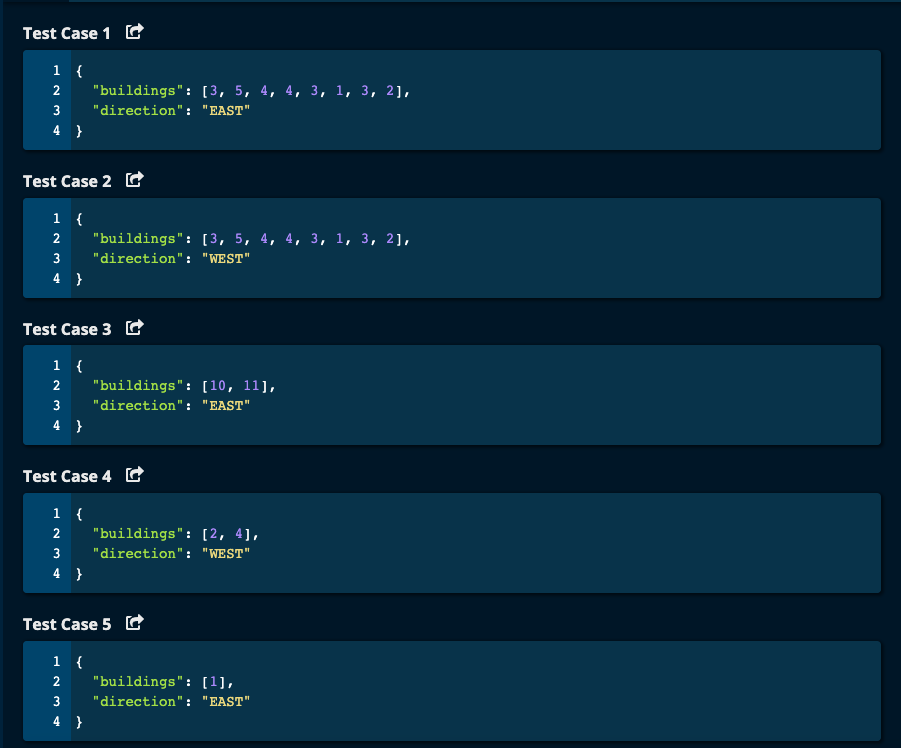
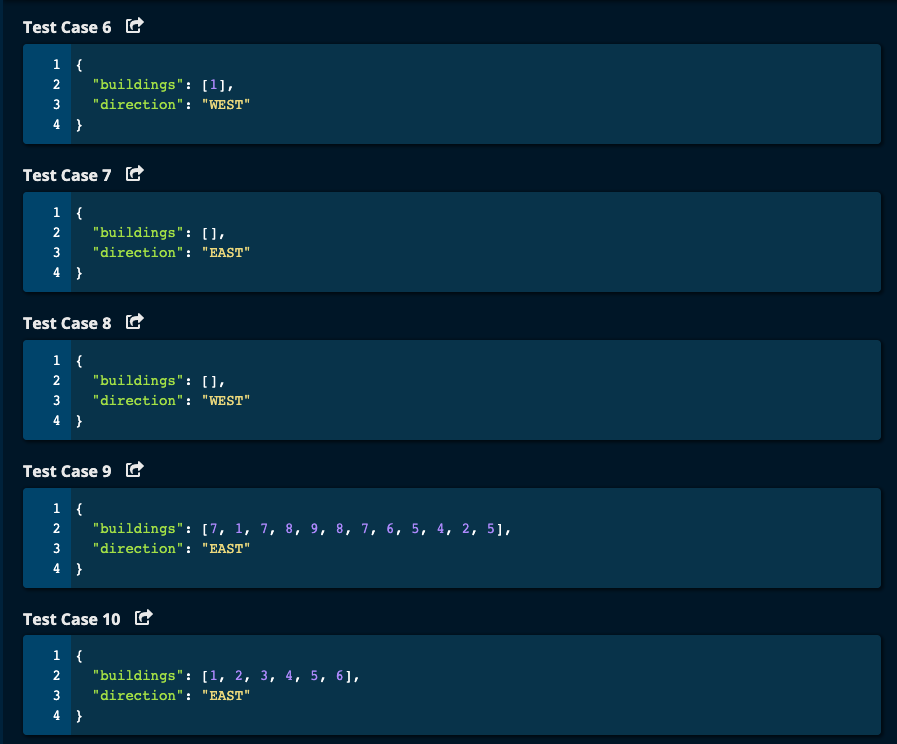
Sunset Views (Medium)

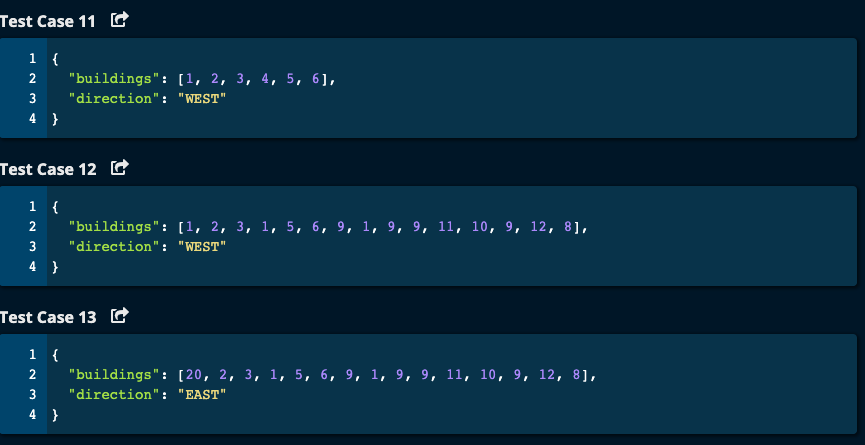












My Solution:

Solution 1: Using max running total. Time Complexity: O(n) | Space Complexity: O(n)

def sunsetViews(buildings, direction):

if len(buildings) == 0:

return buildings

result = []

if direction == "WEST":

start = 0

step = 1

end = len(buildings)

else:

start = len(buildings) - 1

step = -1

end = -1

max = 0 # running maximum building height

for i in range(start, end, step):

if buildings[i] > max:

max = buildings[i]

result.append(i)

if direction == "EAST":

result = result[::-1]

return result

JJ Notes:

1. If length of buildings array is 0, then return buildings array.
2. If the direction is “WEST”, then start is 0, step is 1 and end is length of buildings array.
3. If direction is “EAST, then start is the last element, step is -1 and end is -1/
4. Let the running maximum building height called max be 0.
5. Initialize result array to be an empty list.
6. Iterate through building array. If the current building is more than max, then add its index to the result array.
7. If the direction is “EAST”, then reverse the result array.
8. Return result array.

Solution 2:

# My Solution using Stack -- Time Complexity: O(n) | Space Complexity: O(n)

# empty the stack if the current element is greater than the top of the stack

def sunsetViews(buildings, direction):

if len(buildings) == 0:

return buildings

if direction == "EAST":

start = 0

step = 1

else:

start = len(buildings) - 1

step = -1

stack = []

idx = start

while idx >= 0 and idx <= len(buildings) - 1:

if not stack: # stack is empty

stack.append(idx)

while len(stack) > 0 and buildings[stack[-1]] <= buildings[idx]:

stack.pop()

stack.append(idx)

idx += step

if direction == "WEST":

stack = stack[::-1]

return stack

JJ Notes:

1. If the direction is “EAST”, start = 0 and step = 1
2. If direction is “WEST”, start = end of buildings array and step = -1
3. Initialize the stack as an empty list.
4. Set idx to start.
5. Traverse through the array depending on the direction.
6. If the stack is empty, append idx to the stack.
7. If the stack is not empty and if the top of the stack is less than the current building height, then pop the stack since building in the stack will not be able to see the sunset.
8. Append the current idx to the stack.
9. Apply step to idx.
10. If the direction is “WEST”, then reverse the stack.
11. Return the stack.

Algoexpert Solutions:

Solution 1:



