Arrays Worksheet

# Review Questions

## Gas Station

Refer to this [Gas Station](https://leetcode.com/problems/gas-station/) problem for the following question:

2 of the following statements are true. Devise a proof in plain English for why or why not each statement is true. Then feel free to use those two statements to solve this problem in the worksheet (#11).

* If there is more gas in the stations than total cost to drive. Then there must be a valid solution. True

if gas[i] - cost[i] > 0 for each gas stations, then sum(gas[i] - cost[i]) >= 0, where i is from 0 to n. Under this situation, we have n different solutions.

if sum(gas[i] - cos[i]) >= 0, we can prove that we have at least one gas station as starting point to travel around by using proof by contradiction.

* If there is a valid solution then its best to start from the station that provides the greatest difference in gas and cost. False

we use an example to show this is false.

NO.1 2 3 4 5

gas[1,2,3,4,5]

-cost[3,4,5,2,1]

———————

increment[-2,-2,-2,2,4]

if we start at the last gas station(5), we can only get to gas station 3. This statement is false.

if there is a valid solution then the starting point is the beginning of the maximum subarray. Here subarray is the subarray of the increment array.

* Starting from an arbitrary gas station and begin traversing. If you run out of gas, then start over at the current gas station. Continue until you’ve find a path that works. This would take O(N) runtime to find the right path. False

we use the example of the previous question. If we start at station 1 which its increment is -2, this means we can’t leave at this gas station.

if we start over at the current gas station. We will be stuck in here forever.

* If the circuit starts at some station X and can not reach some station Y. Any station between X and Y  
  can not reach Y. True

suppose increment at a gas station is a[i] = gas[i] - cost[i], then sum(a[i]) < 0 where i is from X to Y.

suppose we have middle gas station Z between X and Y. Z can reach Y(proof by contradiction). So we have sum(a[i]) > 0 where i is from Z to Y.

sum(a[i]) + sum(a[i]) = sum(a[i])

x ->z z -> y x->y

we can conclude sum(a[i]) < 0 where i is from X to Z. But from the original assumption we have sum(a[i]) > 0 where i is from X to Z.

So the statement Z can reach Y is not true. Z can’t reach Y is true.

# Activity Selection Problem:

We have a number of activities with a start time and finish time. We want to know the maximum number of activities that can be performed given that only one activity can be worked on at a time.

We have the following 3 algorithms that solve this problem for the given example. However, only one of the algorithms is correct. Choose the correct algorithm by providing counterexamples to the 2 incorrect algorithms. Then write the code to solve this problem with the correct algorithm. And use the counterexamples you created as test cases to verify your code is correct.

Example: (3,3), (1,6), (6,7), (0,2)

Output: 3 because we can complete tasks: (0,2), (3,3), (6,7)

Algorithm 1:

Sort the tasks by duration and complete the shortest duration tasks first. Keep popping tasks off of this sorted stack. Keep track of all the time intervals we have worked on and then skip tasks that overlap with those time intervals. False

Sorted Tasks: (3,3), (6,7), (0,2),(1,6)

Complete (3,3) then (6,7) then (0,2) but not (1,6) because it overlaps with (3,3) and (0,2)

Example: (0,2),(2,5),(4,6),(5,8)

Complete (0,2) then (4,6) but not (2,5),(5,8)

The expected results:(0,2), (2,5), (5,8)

Algorithms 2:

Sort the tasks by start time and break ties by duration and then keep completing tasks and discard any tasks with overlapping times. False

Sorted Tasks: (0,2), (1,6), (3,3), (6,7)

Complete (0,2) then discard (1,6) for overlap, then complete (3,3) then complete (6,7)

Example: (0, 7), (1,6),(3,3),(6,7)

complete(0,7) then discard the rest, the result is (0,7)

The expected results: (1,6),(6,7) or (3,3),(6,7)

Algorithm 3:

Sort by end time. Pop tasks off the stack and count any task that has a start time greater than any previously selected end time. Correct

Sorted Tasks: (0,2), (3,3),(1,6), (6,7)

Complete (0,2) then complete (3,3), discard (1,6) for overlap then complete (6,7)

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# Tips for Solving Leetcode Problems

* Spend only 45 minutes trying to devise a solution to each problem (everything except coding)
* Please reference our [guide on how to approach practice exercises in Teachable](https://codebreakers1.teachable.com/courses/codebreakers-training-vault/lectures/14591190).

Complete the following Leetcode Problems and add them to your LC Review Schedule.

|  |
| --- |
| 1. [Remove Duplicates From Sorted Array](https://leetcode.com/problems/remove-duplicates-from-sorted-array/) |
| 1. [2SUM](https://leetcode.com/problems/two-sum) |
| 1. [3SUM](https://leetcode.com/problems/3sum) |
| 1. [Longest Mountain in Array](https://leetcode.com/problems/longest-mountain-in-array/) |
| 1. [Lemonade Change](https://leetcode.com/problems/lemonade-change/) |
| 1. [Jump Game](https://leetcode.com/problems/jump-game/solution/) |
| 1. [Best Time to Buy / Sell Stock 2](https://leetcode.com/problems/best-time-to-buy-and-sell-stock-ii/) |
| 1. [Subarray product less than K](https://leetcode.com/problems/subarray-product-less-than-k/) |
| 1. [Longest Substring Without Repeating Characters](https://leetcode.com/problems/longest-substring-without-repeating-characters/) |