

CMPE 282: In-Class Group Assignment: II

Group members:

- Jack Kalavadia (016197519)
- Rutvik Moradiya (016656367)
- Ujwala Balbhim Mote (016711396)

Q1: Use a language model to help your team write a program that does the following:

Include tests showing all the sentiments and submit a word document with screenshots of the program operations and tests.

Include a link to your source code on github, etc..

Use any programming language you prefer.

Here is the code in python 3:

```
import nltk
nltk.download('vader_lexicon')
from nltk.sentiment import SentimentIntensityAnalyzer

def sentiment_analysis(text):
    sentiment = SentimentIntensityAnalyzer().polarity_scores(text)
    if sentiment['compound'] >= 0.05:
        return 'Positive'
    elif sentiment['compound'] <= -0.05:
        return 'Negative'
    else:
        return 'Neutral'

# Test cases
test_texts = [
    'This is an amazing product!',
    'I hate this product',
    'This product is just okay',
]
for text in test_texts:
    print(f'{text}: {sentiment_analysis(text)}')
```

Q2: Use a language model to help your team solve HackerRank Links to an external site.coding problems, in the language of your choice

1.One of: Easy, Medium, or Hard

First, use a simplistic prompt, and show the initial score from HackerRank

Use Prompt Engineering techniques to refine your answer, and show the HackerRank score for the improved answer

Include screenshots of the code operations and results.

Checking, and include a link to your source code on github, etc.

The screenshot shows the HackerRank interface for the problem "1. Unexpected Demand". The problem description states: "A widget manufacturer is facing unexpectedly high demand for its new product. They would like to satisfy as many customers as possible. Given a number of widgets available and a list of customer orders, what is the maximum number of orders the manufacturer can fulfill in full?". The function description requires completing the `filledOrders` function, which takes an array of orders and an integer `k` representing the number of widgets available. The constraints are: $1 \leq n \leq 2 \times 10^5$, $1 \leq \text{order}[i] \leq 10^9$, and $1 \leq k \leq 10^9$. The initial code submission is as follows:

```
1 > #!/bin/python3=
10 #
11 # Complete the 'filledOrders' function below.
12 #
13 # The function is expected to return an INTEGER.
14 # The function accepts following parameters:
15 # 1. INTEGER_ARRAY order
16 # 2. INTEGER k
17 #
18
19 def filledOrders(order, k):
20     total = 0
21     for i, v in enumerate(sorted(order)):
22         if total + v <= k:
23             total += v
24         else:
25             return i
26     else:
27         return len(order)
28
29 > if __name__ == '__main__': =
```

The interface shows a "Test Results" tab with a "Custom Input" section. The "Run Code" button is highlighted, and the "Submit" button is also visible. The "Test Results" section shows a "Compiled successfully. All available test cases passed" message.

The screenshot shows the HackerRank interface for the problem "1. Unexpected Demand" after the code has been submitted. The "Test Results" tab is active, showing a "Compiled successfully. All available test cases passed" message. The "Test Results" section lists the test cases and their status:

Test Case	Status
Test case 0	Passed
Test case 1	Passed
Test case 2	Passed
Test case 3	Passed
Test case 4	Passed
Test case 5	Passed

The "Custom Input" section shows the input for the test cases:

```
Input (stdin)
1 2
2 10
3 30
4 40

Your Output (stdout)
1 2

Expected Output
```

The "Run Code" button is highlighted, and the "Submit" button is also visible. The "Test Results" section shows a "Compiled successfully. All available test cases passed" message.

1h 1m left

2. Maximum Cost of Laptop Count

A company manufactures a fixed number of laptops every day. However, if some defect is encountered during the testing of a laptop, it is labeled as "illegal" and is not counted in the laptop count of the day. Given the cost to manufacture each laptop, its label as "illegal" or "legal", and the number of legal laptops to be manufactured each day, find the maximum cost incurred by the company in a single day in manufacturing all the laptops.

Note that if a laptop is labeled as illegal, its manufacturing cost is still incurred by the company, even though it is not included in the laptop count. Also, days are only taken into when the daily laptop count has been completely met. If there are no such days, the answer is 0.

For example, let's say there are $n = 5$ laptops, where $cost = [2, 5, 3, 11, 1]$. The labels for these laptops are $labels = ["legal", "illegal", "legal", "illegal", "legal"]$. Finally, the $dailyCount = 2$, which means that the company needs to manufacture 2 legal laptops each day. The queue of laptops can be more easily viewed as follows:

- cost 2, "legal"
- cost 5, "illegal"
- cost 3, "legal"
- cost 11, "illegal"
- cost 1, "legal"

On the first day, the first three laptops are manufactured in order to reach the daily count of 2 legal laptops. The cost incurred on this day is $2 + 5 + 3 = 10$. On the second day, the fourth and fifth laptops are manufactured, but because only one of them is legal, the daily count isn't met, so that day is not taken into consideration. Therefore, the maximum cost incurred on a single day is 10.

Function Description

Language Python 3

Autocomplete Ready

1 > #!/bin/python3 -

10 #

11 # Complete the 'maxCost' function below.

12 #

13 # The function is expected to return an INTEGER.

14 # The function accepts following parameters:

15 # 1. INTEGER_ARRAY cost

16 # 2. STRING_ARRAY labels

17 # 3. INTEGER dailyCount

18 #

19

20 def maxCost(cost, labels, dailyCount):

21 # Write your code here

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37 > if __name__ == '__main__':

Test Results Custom Input Run Code Run Tests Submit

Compiled successfully. All available test cases passed

Test case 0

Test case 1

Test case 2

Test case 3

Test case 4

Test case 5

Input (stdin)

1 1

2 2

3 1

4 illegal

5 1

Run as Custom Input Download

Your Output (stdout)

1 0

1h 1m left

2. Maximum Cost of Laptop Count

A company manufactures a fixed number of laptops every day. However, if some defect is encountered during the testing of a laptop, it is labeled as "illegal" and is not counted in the laptop count of the day. Given the cost to manufacture each laptop, its label as "illegal" or "legal", and the number of legal laptops to be manufactured each day, find the maximum cost incurred by the company in a single day in manufacturing all the laptops.

Note that if a laptop is labeled as illegal, its manufacturing cost is still incurred by the company, even though it is not included in the laptop count. Also, days are only taken into when the daily laptop count has been completely met. If there are no such days, the answer is 0.

For example, let's say there are $n = 5$ laptops, where $cost = [2, 5, 3, 11, 1]$. The labels for these laptops are $labels = ["legal", "illegal", "legal", "illegal", "legal"]$. Finally, the $dailyCount = 2$, which means that the company needs to manufacture 2 legal laptops each day. The queue of laptops can be more easily viewed as follows:

- cost 2, "legal"
- cost 5, "illegal"
- cost 3, "legal"
- cost 11, "illegal"
- cost 1, "legal"

On the first day, the first three laptops are manufactured in order to reach the daily count of 2 legal laptops. The cost incurred on this day is $2 + 5 + 3 = 10$. On the second day, the fourth and fifth laptops are manufactured, but because only one of them is legal, the daily count isn't met, so that day is not taken into consideration. Therefore, the maximum cost incurred on a single day is 10.

Function Description

Language Python 3

Autocomplete Ready

1 > #!/bin/python3 -

10 #

11 # Complete the 'maxCost' function below.

12 #

13 # The function is expected to return an INTEGER.

14 # The function accepts following parameters:

15 # 1. INTEGER_ARRAY cost

16 # 2. STRING_ARRAY labels

17 # 3. INTEGER dailyCount

18 #

19

20 def maxCost(cost, labels, dailyCount):

21 # Write your code here

22 maxDailyCost = 0

23 currCount = 0

24 currCost = 0

25 for i in range(len(cost)):

26 if currCount >= dailyCount:

27 maxDailyCost = max(maxDailyCost, currCost)

28 currCount, currCost = 0, 0

29

30 currCost += cost[i]

31 if labels[i] == 'legal':

32 currCount += 1

33

34 print("Laptop", i, "-", cost[i], labels[i], currCount, currCost)

35 print("MaxDailyCost:", maxDailyCost)

36 return maxDailyCost

37 > if __name__ == '__main__':

Test Results Custom Input Run Code Run Tests Submit

Compiled successfully. All available test cases passed

Test case 0

Test case 1

Test case 2

Test case 3

Test case 4

Test case 5

Input (stdin)

1 1

2 2

3 1

4 illegal

5 1

Run as Custom Input Download

Your Output (stdout)

1 0

✓ Test submitted successfully

HackerRank has received your test report. If you would like to contact them, please go to the invitation e-mail you received and select "Reply All".

1. How was your overall experience?

