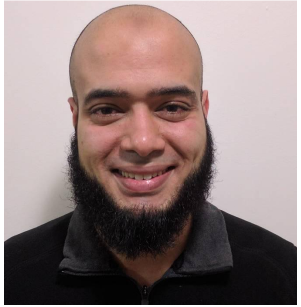
Python Programming

Homework 3

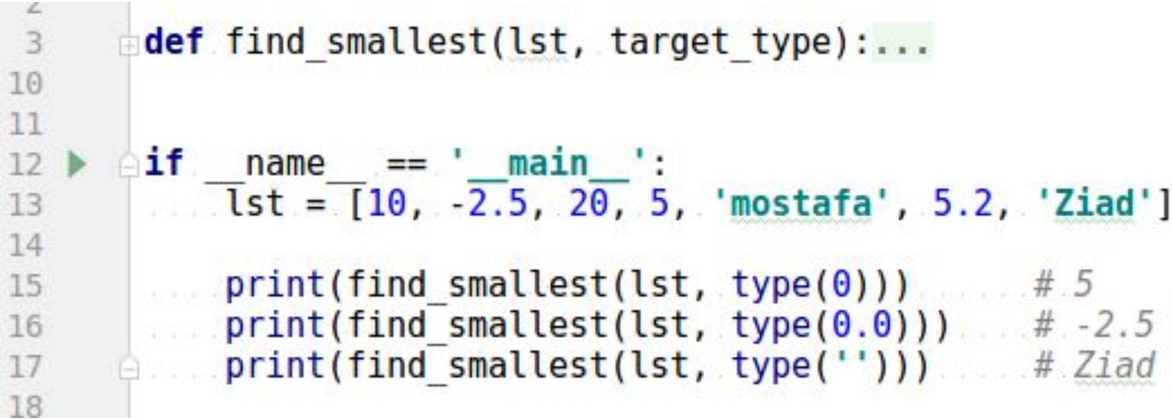
Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)



Homework 1: Minimum of a type!

- Write a function that takes a list and a type
  - It returns the minimum value among this data type or None if not present
  - E.g. In below list: [10, 20, 5] are of type int. Their minimum is 5



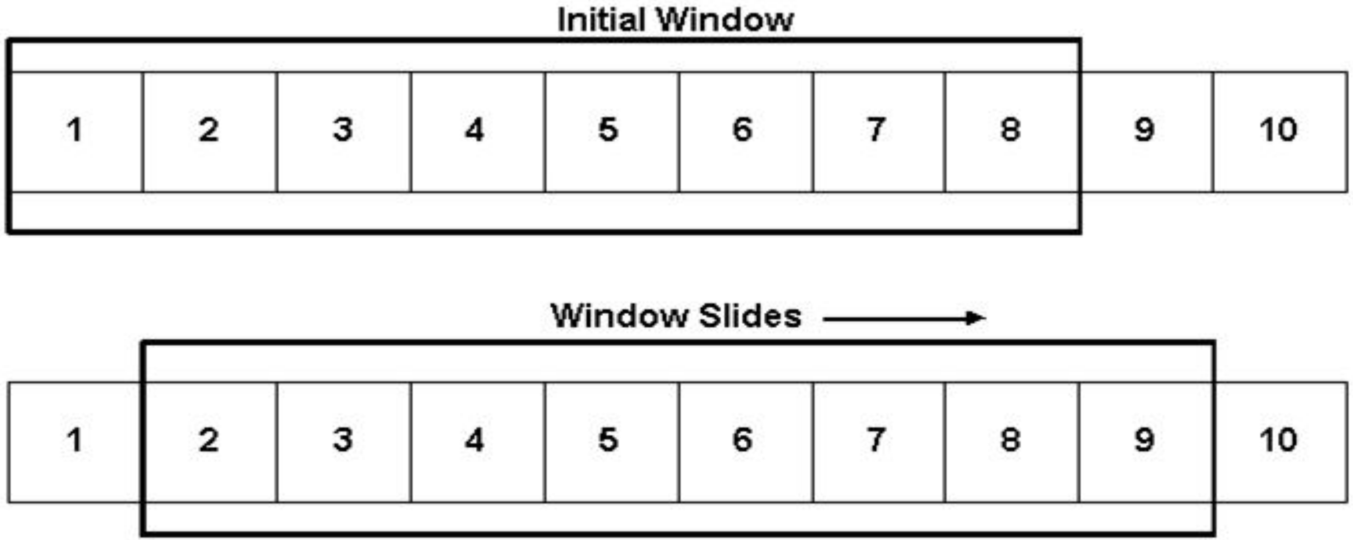
```python
def find_smallest(lst, target_type):...


if __name__ == '__main__':
    lst = [10, -2.5, 20, 5, 'mostafa', 5.2, 'Ziad']

    print(find_smallest(lst, type(0)))      # 5
    print(find_smallest(lst, type(0.0)))    # -2.5
    print(find_smallest(lst, type('')))     # Ziad
```

Homework 3: Is sublist

- Read a line of N integers. Let’s call it list1
- Then Read a line of M integers. Let’s call it list2
- Print YES if list2 is a sublist of list1. Otherwise print NO
  - Sublist is like substring. All of it as it is should appear in list1
- Input ⇒ Output
  - [1 2 3 4]   [1 4]   ⇒ False
  - [1 **2 3** 4]   [2 3]   ⇒ True
  - [10 -10 **20 25 2** 7 2 3]   [20 25 25]   ⇒ True
  - [10 -10 20 25 2 7 2 3]   [20 25 7]   ⇒ False

Background: Fixed Sliding Window

- Indicate a group of consecutive number. Fixed and variable size
  - You slide to next window



Initial Window

1 2 3 4 5 6 7 8 9 10

Window Slides →

1 2 3 4 5 6 7 8 9 10

Img src

Background: Fixed Sliding Window

- Assume a list: **1 0 3** -4 2 -6 9
- Sliding window (sublist): 3
- Let’s print all windows of length 3 and their sum
  - 1 0 3 ⇒ sum = 4
  - 0 3 -4 ⇒ sum = -1 [observe 0 3 are common]
  - 3 -4 2 ⇒ sum = 1
  - -4 2 -6 ⇒ sum = -8
  - 2 -6 9 ⇒ sum = 5
- Observe the relation between 2 consecutive windows:
  - They share all the elements except a change in the first / last element
- Variable sliding window: its size grows and shrinks

Homework 3: Fixed sliding window

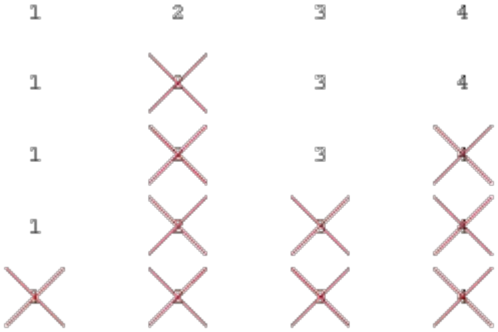
- Read a line of N integers. Then Read integer K (on next line). K <= N
- Find the first sublist of K elements that has maximum sum.
- Input
    - 1 0 3 -4 **2 -6 9**
    - 3
    - Output: Starts at 4 with Sum 5
    - **30 -6 -8 10** 2
    - 4
    - Output: Starts at 0 with Sum 26
- Medium to Hard: Can you do it without nested loops? There are 2 ways.

Homework 4: Count increasing sublists

- Read a line of N integers. Count how many sublist are increasing
- E.g. If input is 1 2 3 4
  - We can find all sublists of length 1 ⇒ [1], [2], [3], [4]
  - All sublists of length 2 ⇒ [1, 2], [2, 3], [3, 4]
  - All sublists of length 3 ⇒ [1, 2, 3], [2, 3, 4]
  - All sublists of length 4 ⇒ [1, 2, 3, 4]
- Inputs ⇒ Outputs
  - 1 2 3 4 ⇒ 10 [10 sublists from previous example, all are increasing]
  - 4 3 2 1 ⇒ 4 [only sublists of length 1 can be considered]
  - 10 20 1 5 ⇒ 6
- Easy using 3 nested loops. Medium using 2 loops. Hard using 1 loop
  - Do your best

Homework 5: Josephus problem

- Read 2 integers N (<= 200) and K (<= 1000000). *Code for small K first*
    - Find the game winner for following game:
- We have a group of N people in Circle. They are numbered 1, 2, …. N
    - Someone is the master of the game.
    - He starts from Person #1. Count K. Then remove this person from the circle.
    - He keeps doing so till only 1 person remains. This is the winner.
- Input 4 2
    - Means we have people: 1, 2, 3, 4. Master starts at 1
    - Count 2 persons (2 removed), start from 3
    - Count 2 persons (4 removed), start from 1
    - Count 2 persons (3 removed), 1 is winner
- Output
    - People removed in order: 2 4 3 1 [same answer for 10 2 why?]



Homework 5: Josephus problem

- Input ⇒ Output
  - 7 1 ⇒ 1 2 3 4 5 6 7
  - 7 2 ⇒ 2 4 6 1 5 3 7
  - 7 3 ⇒ 3 6 2 7 5 1 4
  - 7 4 ⇒ 4 1 6 5 7 3 2
  - 7 5 ⇒ 5 3 2 4 7 1 6
  - 7 6 ⇒ 6 5 7 2 1 4 3
  - 7 7 ⇒ 7 1 3 6 2 4 5
  - 7 14 ⇒ 7 2 6 3 5 4 1
  - 7 1000 ⇒ 6 3 2 1 4 7 5
  - 7 99999 ⇒ 4 7 5 2 1 3 6

Homework 6: longest sublist

- Read a line of N integers. Each is just 0 or 1
- Find the longest **sublist** with number of zeros == numbers of ones
    - Easy: 3 loops
    - Medium: 2 loops (even with no extra arrays)
    - (very) hard: Single loop
- Inputs ⇒ outputs
    - 1 **0 0 0 1 1 1** ⇒ 6 (e.g. 100011 or 000111)
    - 1 0 0 0 **0 0 1 0 1 1 0 1** 0 0 0 0 0 0 1 ⇒ 8 (e.g. 00101101)
    - 1 1 1 1 ⇒ 0
    - 1 1 1 0 0 ⇒ 4
    - 0 ⇒ 0
- Reduction
    - How may this problem be reduced to another problem: longest sublist of zero sum?

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."