

Data Structure and Programming, Spring 2024  
programming assignment #5  
**Top-k Sum of Subarray**

Shao-Ming, Chen                      Chiao-Yun, Chin  
r11942095@ntu.edu.tw              b10901154@ntu.edu.tw

Pei-Yuan, Wu  
peiyuanwu@ntu.edu.tw

RELEASE DATE: 05/23/2024  
DUE DATE: 06/06/2024

## 1 Problem Description

Given an array composed of  $n$  integers and a number  $k \leq n$ , find the top  $k$ -th largest sum of a **contiguous** subarray within the array, considering the presence of both positive and negative numbers. e.g. for array = [3, 1, -1]:

- Top-1 sum of subarray = 4. (subarray = [3, 1])
- Top-2 sum of subarray = 4, 3. (subarray = [3, 1], [3] or [3, 1], [3, 1, -1])
- Top-3 sum of subarray = 4, 3, 3. (subarray = [3, 1], [3], [3, 1, -1])

## 2 Input/Output Specification

We have 5 test cases in total, and we provide 3 of these test cases: input\_1.json, input\_2.json, input\_3.json, and their corresponding golden output: golden\_1.json, golden\_2.json, golden\_3.json. *Note: We have finished I/O handling in **main.py**.*

### 2.1 Input format

The input file is a json format file with two fields: "array" and "topk", indicating what array you should compute for the top- $k$  values of sum of subarray.

## 2.2 Output format

The output file is a json format file with a sequence of  $k$  numbers, indicating the top- $k$  values of sum of subarray w.r.t. the given array. Note that the resulting subarray of corresponding top- $k$  value should NOT be returned.

## 3 Evaluation

We provide the template: ***main.py*** for input/output handling. Your task is to complete the sections marked as **TODO**.

1. **The time complexity of your code should not exceed  $O(nk \log n)$ .**  
As a reference, the runtime of input\_3.json is around 0.025s. Remember that it is a reference value and we will grade based on your code if your code runtime exceeds the number too much.
2. You are free to use any algorithm you prefer, e.g. brute-force, divide and conquer, heap data structure, etc. However, since we have set time limit constraint, it is recommended to consider efficiency of your method.
3. Use command "bash evaluation.sh" to check your code.
4. Do NOT modify the interface of the functions, but you can add your own functions.
5. All *python standard libraries* (except *heapq*) are allowed. Other libraries such as *numpy* are NOT allowed. If you have any question regarding library permission, you are welcome to email TA.
6. Environment: Python 3.10, Ubuntu 20.04.

## 4 Grading policy

This programming assignment will be graded based on correctness and time complexity.

1. The total point is 100.
2. If your program passes the test cases, for each case you will get 10%.
3. If your program passes the test cases with desired time complexity, for each case you will get additional 10%.
4. **Don't plagiarize.**

## 5 Submission

Please put *main.py* into a directory named **studentID** and compress the directory into **studentID.zip** to NTU COOL. The homework is due on **06/06**, at **23:59**

**Note: If the file structure of your attachment is not correct, you will be deducted 10%.**