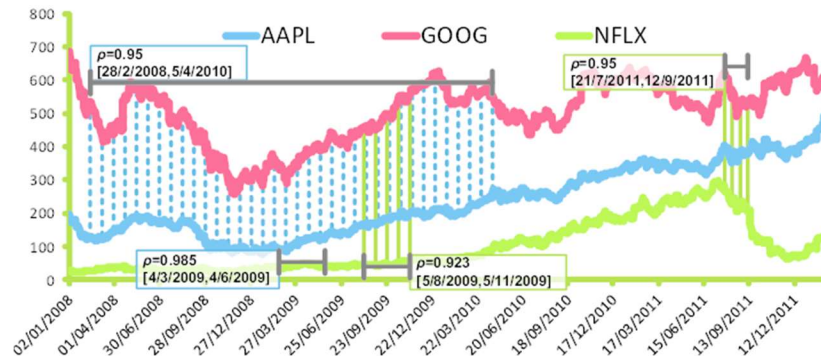


F. Financial Data Analysis

From 1 to n ($n \leq 10^6$)

Time Limit: 3 seconds

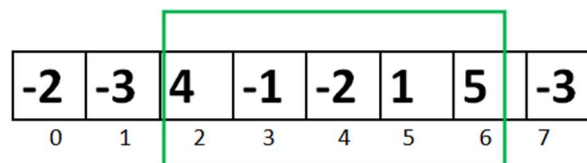


Problem description

In the ever-changing landscape of finance, the analysis of stock prices and financial time series data stands as a pivotal endeavor. Financial professionals, including investors, traders, and analysts, face the challenge of deciphering intricate patterns within these datasets to facilitate well-informed decision-making. One particularly crucial facet of this analytical process involves the utilization of the "Largest Sum Contiguous Subarray" algorithm.

This algorithm plays a pivotal role in the domain of finance, where it is employed to extract meaningful insights from stock prices and financial time series data. By identifying contiguous subarrays with the highest cumulative sums, it reveals critical time periods characterized by distinctive market behaviors. It is within these periods that investors can seize opportunities to maximize profits or mitigate losses, rendering the largest sum contiguous subarray algorithm an indispensable tool for guiding financial decisions.

Largest Subarray Sum Problem



$$4 + (-1) + (-2) + 1 + 5 = 7$$

Maximum Contiguous Array Sum is 7

INPUT	OUTPUT
First line: number n indicates the quantity of rows containing the values of stock prices in	First line: store index of the elements which included in the largest sum contiguous subarray

time series which can be positive or negative value. $n \leq 10^6$ Next n lines: store value of stock price, where value $\leq 10^6$	Second line: maximum contiguous array sum value
---	---

Example 1 (from the above figure):

INPUT	OUTPUT
8 -2 -3 4 -1 -2 1 5 -3	2 3 4 5 6 7

Example 2:

INPUT	OUTPUT
10 33098 52605 9471 -52596 19628 -66768 -42601 -4815 36417 78407	8 9 114824

Where in output:

element 8 has value **36417**

element 9 has value **78407**

and **sum** = **36417** + **78407** = **114824**