

תרגיל 5:

1. יצרנו קובץ בשם profiling.cpp
2. בתוך הקובץ, כתבנו את שלושת הפונק' הנדרשות לכל אחת מהסיבוכיות שנדרשו:
 - א. maxSubarraySumKadane בסיבוכיות $O(n)$
 - ב. maxSubarraySumQuadratic בסיבוכיות $O(n^2)$
 - ג. maxSubarraySumCubic בסיבוכיות $O(n^3)$ובנוסף פונקציית עזר שמוציאה לנו את המספרים מתוך seed ומחזירה אותם במודולו 100 ומפחיתה 25 על מנת להיות בטווח הרצוי.

הוראות קימפול:

```
nitzan@nitzan-VirtualBox:~/Desktop/OS_EX1/q5$ make  
g++ -Wall -Wextra -O2 -pg -o profiling profiling.cpp
```

הוראות הרצה:

`./profiling <seed> <generated_numbers>`

דוגמת הרצה:

יצרנו 100 מספרים:

```
nitzan@nitzan-VirtualBox:~/Desktop/OS_EX1/q5$ ./profiling 42 100  
Running Kadane's Algorithm ( $O(n)$ )...  
Result:2075  
Running Quadratic Algorithm ( $O(n^2)$ )...  
Result:2075  
Running Cubic Algorithm ( $O(n^3)$ )...  
Result:2075  
All algorithms executed successfully!
```

יצרנו 1000 מספרים:

```
nitzan@nitzan-VirtualBox:~/Desktop/OS_EX1/q5$ ./profiling 42 1000  
Running Kadane's Algorithm ( $O(n)$ )...  
Result:25530  
Running Quadratic Algorithm ( $O(n^2)$ )...  
Result:25530  
Running Cubic Algorithm ( $O(n^3)$ )...  
Result:25530  
All algorithms executed successfully!
```

יצרנו 10000 מספרים:

```
nitzan@nitzan-VirtualBox:~/Desktop/OS_EX1/q5$ ./profiling 42 10000
Running Kadane's Algorithm (O(n))...
Result:247358
Running Quadratic Algorithm (O(n^2))...
Result:247358
Running Cubic Algorithm (O(n^3))...
Result:247358
All algorithms executed successfully!
```

לכל אחת מהריצות שביצענו, יצרנו קובץ gprof 100XX ייחודי לה:



יצירת הקבצים הללו התבצעה על ידי:

```
nitzan@nitzan-VirtualBox:~/Desktop/OS_EX1/q5$ gprof profiling gmon.out > gprof_100.txt
```

נבחן את המידע בקבצים:

:gprof_100

```
Each sample counts as 0.01 seconds.
no time accumulated
```

% time	cumulative seconds	self seconds	calls	self Ts/call	total Ts/call	name
0.00	0.00	0.00	3	0.00	0.00	frame_dummy
0.00	0.00	0.00	1	0.00	0.00	generateRandomInput(int, int)
0.00	0.00	0.00	1	0.00	0.00	maxSubarraySumCubic(std::vector<int, std::allocator<int> > const&)
0.00	0.00	0.00	1	0.00	0.00	maxSubarraySumKadane(std::vector<int, std::allocator<int> > const&)
0.00	0.00	0.00	1	0.00	0.00	maxSubarraySumQuadratic(std::vector<int, std::allocator<int> > const&)
0.00	0.00	0.00	1	0.00	0.00	std::_Vector_base<int, std::allocator<int> >::~~Vector_base()

:gprof_1000

```
Each sample counts as 0.01 seconds.
no time accumulated
```

% time	cumulative seconds	self seconds	calls	self Ts/call	total Ts/call	name
0.00	0.00	0.00	3	0.00	0.00	frame_dummy
0.00	0.00	0.00	1	0.00	0.00	generateRandomInput(int, int)
0.00	0.00	0.00	1	0.00	0.00	maxSubarraySumCubic(std::vector<int, std::allocator<int> > const&)
0.00	0.00	0.00	1	0.00	0.00	maxSubarraySumKadane(std::vector<int, std::allocator<int> > const&)
0.00	0.00	0.00	1	0.00	0.00	maxSubarraySumQuadratic(std::vector<int, std::allocator<int> > const&)
0.00	0.00	0.00	1	0.00	0.00	std::_Vector_base<int, std::allocator<int> >::~~Vector_base()

:gprof_10000

Each sample counts as 0.01 seconds.

% time	cumulative seconds	self seconds	calls	self s/call	total s/call	name
99.62	2.64	2.64	1	2.64	2.64	maxSubarraySumCubic(std::vector<int, std::allocator<int> > const&)
0.38	2.65	0.01	1	0.01	0.01	maxSubarraySumQuadratic(std::vector<int, std::allocator<int> > const&)
0.00	2.65	0.00	3	0.00	0.00	frame_dummy
0.00	2.65	0.00	1	0.00	0.00	generateRandomInput(int, int)
0.00	2.65	0.00	1	0.00	0.00	maxSubarraySumKadane(std::vector<int, std::allocator<int> > const&)
0.00	2.65	0.00	1	0.00	0.00	std::_Vector_base<int, std::allocator<int> >::~~_Vector_base()

ניתן לראות , שבפלט האחרון, שדרש הבאת 10,000 מספרים , הזמן היה ממושך לעומת 100 או 1000 שדרשנו פעמים קודמות.

כמו כן , ניתן לשים לב בבירור להבדל בין זמני הריצה בתוך ההרצה הזו בין הפונקציות עם הסיבוכיות השונות.