

PART 2 - Profiling

ביצענו Profiling באופן הבא -

הכנו קובץ test.txt שבו מספר הקודקודים הינו 20. ולאחר מכן יש הזנה "אוטומטית" של 20 קודקודים עם ערכים שונים על מנת לקבל הבדל ניכר בעת הפעלת שני האלגוריתמים השונים בחישוב שטח הקמור.

שטח הקמור חושב באמצעות שני Data structures -

Vector.1

Deque.2

את חישוב האלגוריתם ביצענו כ-50000 פעם (בתוך לולאה וקריאות לשני הפונקציות)

ניתן לראות מהצילום מסך שזהו פלט ה-PROFILING בקובץ TXT

שאכן יש הבדל בין שני ה-Data structures

השימוש בDeque לקח את אחוז השימוש הגבוה מביניהם (כ-4.27 אחוזים)

לעומת ה-Vector שלקח 2.56 אחוזים

בנוסף יש הבדל בזמן הכולל בתוכנית -

עבור Deque - רץ כ-4.27 אחוז מסך ריצת כל התכנית

לעומת Vector - שרץ כ-2.56 אחוז.

קל לראות ש-Vector אכן יותר יעיל. הוא מאחסן את האיברים בזיכרון באופן רציף.

כלומר, יש גישה מהירה יותר וכן השימוש באיטרטורים פשוט יותר. לעומת Deque

שומר את הנתונים בחתיכות באופן מפוזר בזיכרון, ולא ברצף. ולכן להגיע לכל איבר

נדרש לדלג בין כל חתיכה ולכן לוקח יותר זמן.

```
1 20
2 0,2
3 10,3
4 10,10
5 5,5
6 4,1
7 0,0
8 1,1
9 2,2
10 3,3
11 23,544
12 50,0
13 0,50
14 6,9
15 100,0
16 1056,50
17 50,654
18 0,100
19 34,50
20 22,34
21 14,87
```

```
Each sample counts as 0.01 seconds.
% cumulative self self total
time seconds seconds calls ms/call ms/call name
8.55 0.20 0.20 38000000 0.00 0.00 std::pair<std::__strip_reference_wrapper<std::decay<float&>::type>::__type, std::__strip_reference_wrapper<st
6.20 0.34 0.14 211500162 0.00 0.00 __gnu_cxx::__normal_iterator<Point*, std::vector<Point, std::allocator<Point> > >::operator*() const
5.77 0.48 0.14 104000008 0.00 0.00 std::Deque_iterator<Point, Point&, Point*>::Deque_iterator(std::Deque_iterator<Point, Point&, Point*> con
5.13 0.60 0.12 152000000 0.00 0.00 float& std::forward<float&>(std::remove_reference<float&>::type&)
5.13 0.72 0.12 94000034 0.00 0.00 Convex::orientation(Point, Point, Point)
4.27 0.82 0.10 500000 0.00 0.00 Convex::findConvexHull_using_deque()
3.63 0.91 0.09 31500017 0.00 0.00 Convex::findConvexHull_using_deque()::{lambda(Point const&, Point const&)#1}::operator()(Point const&, Point
2.56 0.96 0.06 48999990 0.00 0.00 __gnu_cxx::__normal_iterator<Point*, std::vector<Point, std::allocator<Point> > >::operator++()
2.56 1.02 0.06 31500017 0.00 0.00 Convex::findConvexHull_using_vector()::{lambda(Point const&, Point const&)#1}::operator()(Point const&, Point
2.56 1.08 0.06 31000000 0.00 0.00 std::reverse_iterator<std::Deque_iterator<Point, Point&, Point*> >::reverse_iterator(std::Deque_iterator<Po
2.56 1.15 0.06 500000 0.00 0.00 Convex::findConvexHull_using_vector()
```