

Algorithm analysis

Dijkstra algorithm to find the fastest path of a set of blocks

Name: Jonathan D. Freire V.

02/11/2018

DESCRIPTION OF THE ALGORITHM

What this algorithm does is create a set of n^2 blocks entered by the user, then the user must enter a starting point (any of the intersections) to finally calculate the fastest way to go to the rest of the other blocks.

EXAMPLE

Suppose that you have to analyze a set of 9 blocks of a city in order to find the better way to get from one place to another.



1) So first you have to put the size of the area (blocks) that you will cover, in this is an area of 3X3, so you put 3.

```

Dell@DESKTOP-T5ST9D9 ~
$ ./a
Ingrese el tamaño de las cuadras
3
    
```

2) Then we have to put the starting point. In this case we will put the point 0.

```

Dell@DESKTOP-T5ST9D9 ~
$ ./a
Ingrese el tamaño de las cuadras
3
Punto de Inicio: 0
    
```

3) As a result we will obtain which is the best path to go from the starting point to the rest of the points of the map.

```

$ ./a
Ingrese el tamaño de las cuadras
3

Punto de Inicio: 0
Punto de Inicio | Punto de llegada | Recorr. Total | Recorrido
0               1               1             1 <-0
0               2               2             2 <-1<-0
0               3               1             3 <-0
0               4               2             4 <-1<-0
0               5               3             5 <-2<-1<-0
0               6               2             6 <-3<-0
0               7               3             7 <-4<-1<-0
0               8               4             8 <-5<-2<-1<-0

De11@DESKTOP-T5ST9D9 ~
$

```

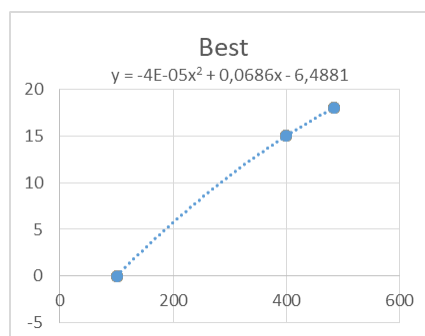
EXECUTION TIME

For the execution time we are going to evaluate 2 cases: Best and Worst Cases.

For the best case we are going to find all the points from the middle of the adjacency matrix of the graph, while for the worst case our starting point is going to be the first node (the paths are going to be largest)

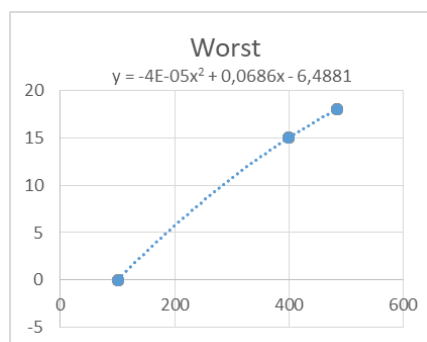
Best Case

Cant. Of nodes	time (milliseconds)
100	0
400	15
484	18



Worst Case

Cant. Of nodes	time (milliseconds)
100	0
400	15
484	18



COMPLEXITY

- In both cases the complexity is $O(n^2)$

CONCLUSION

- There were a problem measuring the complexity of the algorithm because the maximum number of nodes allowed was 484 and this produced that only 3 data can be measured, having a bad regression.
- The best a worst cases have the same values because we have only 3 data.
- Dijkstra algorithm can find the best path in a very good way and can be used for a lot of purposes, not only for maps.