# **Laboratory practice No. 5:**

# **Divide to Conquer and Dynamic Programming**

|  |  |
| --- | --- |
| **David Alzate Cardona**  Universidad Eafit  Medellín, Colombia  Dalzatec1@eafit.edu.co | **Laura Alzate Madrid**  Universidad Eafit  Medellín, Colombia  lalzatem@eafit.edu.co |

**3) Practice for final project defense presentation**

**3.3** The data structure used to solve this problem was an arraylist composed by pairs. The algorithm uses its own method to compare doubles, which is in charge of taking the starting point, it means, the spot where the robot is located, and then it compares which is the closest point according to the data related to the radioactive garbage, then when the algorithm knows which one is the fewer, it returns it, and it moves one square, it could be in a horizontal way or a vertical way and each movement has a cost of 1. To be able to find the minimum distance, we use an algorithm to find the hypotenuse, it's known as Phytagoras..

**3.4** The question is repeated in the document.

**3.5** the order of the algorithm (without taking into account when looking for the closest current point)O(N\*K)

**3.6**

n is the number of maps

K is the number of waste

***4) Practice for midterms***



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | ε | c | a | l | l | e |
| ε | 0 | 1 | 2 | 3 | 4 | 5 |
| c | 1 | 0 | 1 | 2 | 3 | 4 |
| a | 2 | 1 | 0 | 1 | 2 | 3 |
| s | 3 | 2 | 1 | 1 | 2 | 3 |
| a | 4 | 3 | 2 | 2 | 2 | 3 |



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | ε | m | a | d | r | e |
| ε | 0 | 1 | 2 | 3 | 4 | 5 |
| m | 1 | 0 | 1 | 2 | 3 | 4 |
| a | 2 | 1 | 0 | 1 | 2 | 3 |
| m | 3 | 2 | 1 | 1 | 2 | 3 |
| a | 4 | 3 | 2 | 2 | 2 | 3 |

*4.2*

* + 1. O (length x\* length y)
    2. return table [lenx][leny];
  1. 1. a) O (n)
     2. *c)* T (n) = c12n + c2
  2. c)
  3. 1. c) T(n)=T(n/2)+C que es O(log n)
     2. **a**[mitad]
     3. a, mitad + 1, de, z
  4. 1. scm[i]=0
     2. scm[i]=arr[j];
     3. max++;
     4. c) O(n2)
  5. 1. int ni = g[i][j];
     2. int nj = g[k][j];
     3. int nk = g[i][k];
     4. O(n3)