Complexity Documentation

Main-searchFloorArrive Algorithm

|  |  |
| --- | --- |
|  |  |
| int[][]building=new int[numFloors][office]; | 1 |
| int numberOffice=numFloors\*office; | 1 |
| Int floor=0; | 1 |
| for(int i=0;i<numFloors; i++) { | n+1 |
| for(int j=0;j<office;j++) { | n\*m+1 |
| building[i][j]=numberOffice; | n\*m |
| numberOffice--; | n\*m |
| for(int i=0;i<numFloors; i++) { | n+1 |
| for(int j=0;j<office;j++) { | n\*m+1 |
| if(building[i][j]==officeB) { | n\*m |
| floor=i+1; | 1 |
| return floor; | 1 |

Time Complexity = =

|  |  |  |
| --- | --- | --- |
| **Tipo** | **Variable** | **Valores atómicos** |
| Entrada | numFloors  office  officeB | 1  1  1 |
| Auxiliar | Builiding  numberoffice | n\*m  1 |
| Salida | floor | 1 |

Space Complexity =

Max\_PriorityQueue-swap Algorithm

|  |  |
| --- | --- |
|  |  |
| E temp=get(i); | 1 |
| Node[i]=get(j); | 1 |
| Node[j]=temp; | 1 |

Time Complexity

|  |  |  |
| --- | --- | --- |
| **Tipo** | **Variable** | **Valores atómicos** |
| Entrada | i  j | 1  1 |
| Auxiliar | temp  Node[i]  Node[j] | 1  1  1 |
| Salida |  | 0 |

Space Complexity