|  |  |  |  |
| --- | --- | --- | --- |
| Tipo | Variable y llamados | Tamaño 1 valor atómico | Cantidad valores atómicos |
| Entrada | P | (pisoACT + oficina = 8 bytes) + (nombre = 2bytes) | (pisoACT+oficina=2)+(nombre=n) |
|  | Oficinas | 4bytes | 1 |
| Auxiliar | p.getOficina | 4bytes | 1 |
|  | edificio.containsKey(p.getOficina() | 4bytes | 1 |
| Salida | p.toString V p.toString2 V \*return\* | [(nombre = 1byte) + (resto de la cadena de texto = 36bytes)] V [(nombre = 2bytes)+(resto de la cadena de texto=34bytes] V [(nombre=1byte)+(resto de la cadena=138bytes)] | (nombre = n) + 1 |

Complejidad espacial algoritmo ascensor

Al sumar ordenadamente los valores de la columna Cantidad de valores atómicos obtenemos:

*f(n)* = 6+2n

La función usando la notación asintótica big O viene siendo debido a que es un grado mayor a *f(n)*

Complejidad espacial algoritmo print

|  |  |  |  |
| --- | --- | --- | --- |
| Tipo | Variable y llamados | Tamaño atómico 1 valor | Cantidad valores atómicos |
| Entrada | C | 4bytes | 1 |
| Auxiliar | I | 4bytes | 1 |
| Salida | Out | (cadena predeterminada=4bytes)+(edificio.get(i)=2bytes) | (Cadena predeterminada = 1+(n-1)) + n |

Al sumar ordenadamente los valores de la columna Cantidad de valores atómicos obtenemos:

*f(n)* = 3+n

La función usando la notación asintótica big O viene siendo debido a que es un grado mayor a *f(n)*