



Escuela de computación
Ingeniería en computación
IC-3002 Análisis de algoritmos

Quiz

Número 4 y 5

Brandon Retana Chacón
sthvar@estudiantec.cr
2021121141

Alajuela, Costa Rica
Marzo 2022

1. Costo

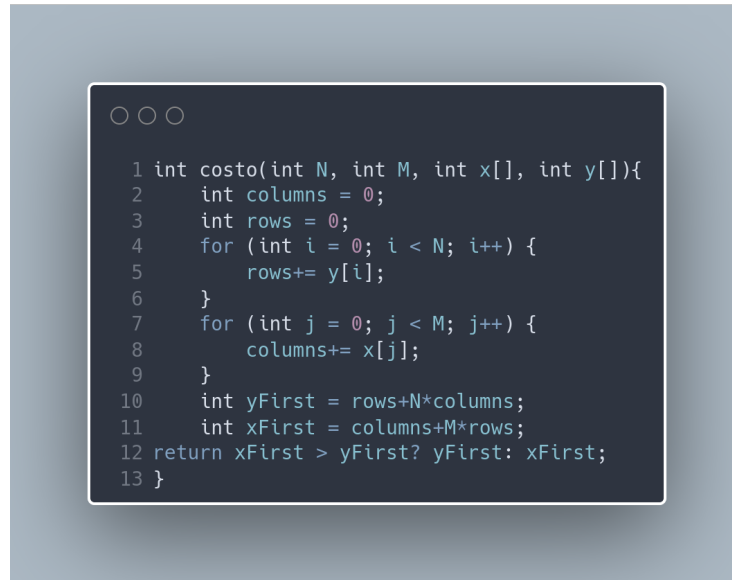


Figura 1: Costo

Función:

$$f(n) = 6n + 6m + 12$$

O grande:

$$O(n + k)$$

Loop invariant:


En la línea 4 y 7.

Las condiciones de estos loops invariant son:

$$\sum_{i=0}^{N-1} y_i \geq rows$$

$$\sum_{i=0}^{M-1} x_i \geq columns$$

2. RadixSort



```
1 void radixSort(int *arr, int n, int max) {
2     int i, j, m, p = 1, index, temp, count = 0;
3     list<int> pocket[10]; //radix of decimal number is 10
4     for(i = 0; i < max; i++) {
5         m = pow(10, i+1);
6         p = pow(10, i);
7         for(j = 0; j < n; j++) {
8             temp = arr[j] % m;
9             index = temp / p; //find index for pocket array
10            pocket[index].push_back(arr[j]);
11        }
12        count = 0;
13        for(j = 0; j < 10; j++) {
14            //delete from linked lists and store to array
15            while(!pocket[j].empty()) {
16                arr[count] = *(pocket[j].begin());
17                pocket[j].erase(pocket[j].begin());
18                count++;
19            }
20        }
21    }
22 }
```

Figura 2: RadixSort

Función:

$$f(n) = 47m + 33mn + 8$$

O grande:

$$O(nk)$$

Loop invariant:

En la línea 12.

La condición de este loop invariant es:

$$count \leq N$$