



PRÁCTICA 1: EFICIENCIA

Ejercicio 4: Mejor y peor caso



Hardware

CPU: Intel Core i7-7700HQ @ 2.80 GHz 8 cores, Intel HD Graphics 630

RAM: 8GB

Sistema operativo: Ubuntu 18.04 LTS 64-bit

Compilador: g++ versión 7.4.0

Opciones de compilación: -o

Para el estudio del mejor caso hemos creado una función que llena el vector con números ordenados de la siguiente forma:

```
void crearMejorCaso(int *v, int tam)
{
    for(int i=0; i<tam; i++)
        v[i] = i;
}
```

Por otro lado, para el estudio del peor caso posible se ha utilizado el siguiente método, el cual crea un vector ordenado en orden inverso:

```
void crearPeorCaso(int *v, int tam)
{
    for(int i=0; i<tam; i++)
        v[i] = tam - i;
}
```

Como podemos observar en la siguiente gráfica, el tiempo del mejor caso posible es mejor que el caso promedio. Lo extraño, es que también el tiempo para el peor caso posible es mejor que el caso promedio, cuando supuestamente debería tardar más que el caso promedio.

