

Práctica 1. Análisis de Eficiencia de Algoritmos

Algorítmica

Shao Jie Hu Chen Mario Megías Mateo Jesús Samuel García Carballo

Universidad de Granada

31 de marzo de 2022

Índice de contenidos

- 1 Introducción
- 2 Eficiencia teórica
- 3 Conclusiones
- 4 Eficiencia empírica
- 5 Referencias

Objetivos

Los objetivos de esta práctica son los siguientes:

- **Estudio teórico, empírico e híbrido y comparación** de los algoritmos de ordenación más empleados, verificando los resultados teóricos.
- **Estudio teórico, empírico e híbrido** de algoritmos de alta complejidad, poniendo especial énfasis en su **viabilidad** en diferentes equipos.
- Estudio del **aumento de eficiencia** de un mismo algoritmo para diferentes **tipos de optimización** del compilador.
- Determinación del algoritmo **más adecuado** para cada situación en función del estado de los datos.

Equipo

ASUS

- **Modelo:** ZenBook 15 UX534F
- **Procesador:** Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz5
- **Memoria Ram:** 16 GB DDR4 @ 2.133 MHz.
- **Sistema Operativo** Ubuntu 20.04.2 LTS

HP

- **Modelo:** Pavilion Gaming Laptop 15-dk0xxx
- **Procesador:** Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz
- **Memoria RAM:** 32 GB DDR4
- **Sistema Operativo:** Ubuntu 20.04.4 LTS

LENOVO

- **Modelo:** YOGA 530-14IKB
- **Procesador:** Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz
- **Memoria RAM:** 8 GB DDR4
- **Sistema Operativo:** Ubuntu 20.04.4 LTS

GraphKiller

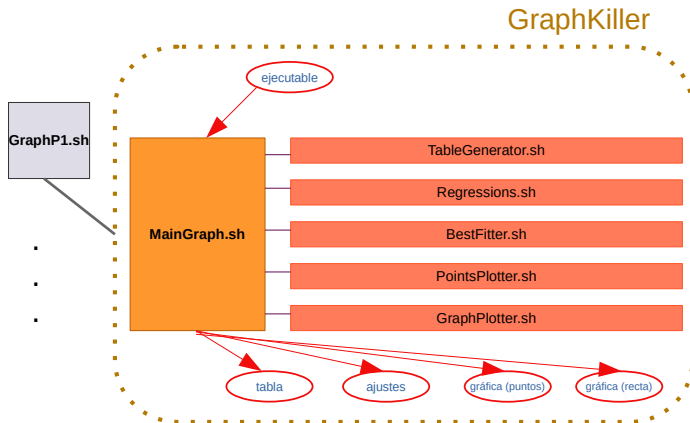


Figura: Esquema del funcionamiento de GraphKiller

Algoritmo HeapSort




Ejemplo de código

```
1 static void heapsort(int T[], int num_elem)
2 {
3     int i;
4     for (i = num_elem/2; i >= 0; i--)
5         reajustar(T, num_elem, i);
6     for (i = num_elem - 1; i >= 1; i--)
7     {
8         int aux = T[0];
9         T[0] = T[i];
10        T[i] = aux;
11        reajustar(T, i, 0);
12    }
13 }
```

Ejemplo de código

```
1 static void heapsort(int T[], int num_elem)
2 {
3     int i;
4     for (i = num_elem/2; i >= 0; i--)
5         reajustar(T, num_elem, i);
6     for (i = num_elem - 1; i >= 1; i--)
7     {
8         int aux = T[0];
9         T[0] = T[i];
10        T[i] = aux;
11        reajustar(T, i, 0);
12    }
13 }
```

Algoritmo de Inserción

-  Verdegay Galdeano. (2017). Lecciones de Algorítmica / José Luis Verdegay. Técnica Avicam.
-  Cormen. (2017). Introduction to algorithms / Thomas H. Cormen... [et al.] (3rd ed.). PHI Learning.
-  Garrido Carrillo. (2018). Estructuras de datos avanzadas: con soluciones en C++ / A. Garrido. Universidad de Granada.