

**Autores:** Francisco Javier Marqués Gaona, Germán Ruano García y David Subires Parra

## **Ejercicio 0 Tema 3**

### ***Ejercicio 0.***

0. Aplicar Dijkstra para evaluar la mejor localización de un Centro Logístico, aquella ubicación en que el coste de distribuir los productos es mínimo. Dijkstra nos da los caminos mínimos para alcanzar todas las otras ubicaciones desde un punto dado; nosotros deseamos encontrar el punto desde dónde es mejor realizar la distribución. Se ha de:
- Presentar el esquema algorítmico detallado (pseudocódigo).
  - Seleccionar las estructuras de datos adecuadas para almacenar los datos.
  - Implementar el correspondiente código en Java.
  - Evaluar experimentalmente la eficiencia utilizando diversos juegos de prueba generados aleatoriamente (dado el tamaño del problema).
  - Almacenar juegos de prueba, resultados y tiempos de ejecución.
  - Analizar la eficiencia obtenida empíricamente frente a la teórica.

---

#### **Pseudocódigo mejorLocalizacion**

```
mejorLocalizacion(){
```

```
    minimo = INFINITO
```

```
    para v ∈ Vertices hacer
```

```
        distancias[] = dijkstra(v,Grafo)
```

```
        sumaDistancia = 0
```

```
        para cada entero ∈ distancias[] hacer
```

```
            sumaDistancia = sumaDistancia + distancia[i]
```

```
        si suma < mínimo
```

```
            minimo = suma
```

```
            mejorVertice = v
```

```
    devolver mejorVertice
```

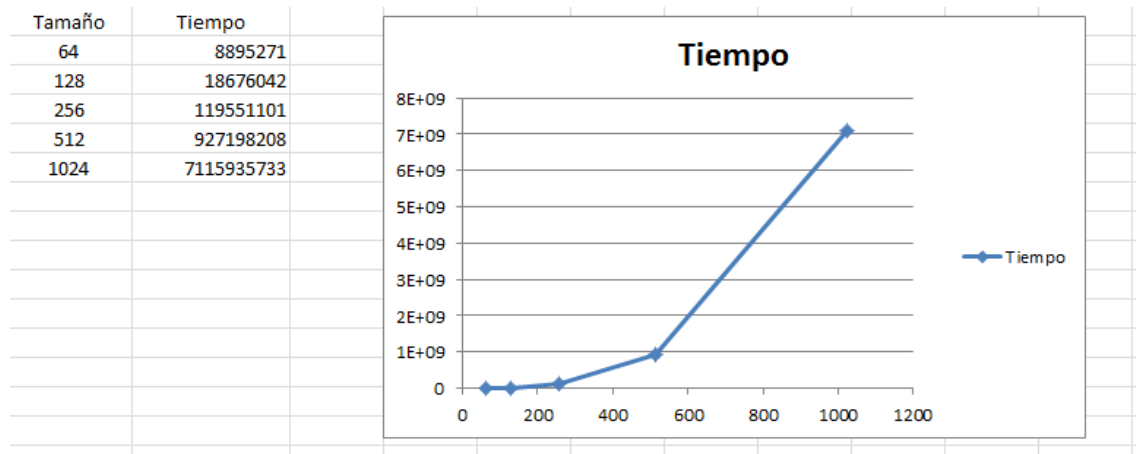
```
}
```

### Pseudocódigo Dijkstra

```
DIJKSTRA (Grafo G, nodo_fuente s)
  para u ∈ V[G] hacer
    distancia[u] = INFINITO
    padre[u] = NULL
  distancia[s] = 0
  adicionar (cola, (s, distancia[s]))
  mientras que cola no es vacía hacer
    u = extraer_mínimo(cola)
    para todos v ∈ adyacencia[u] hacer
      si distancia[v] > distancia[u] + peso (u, v) hacer
        distancia[v] = distancia[u] + peso (u, v)
        padre[v] = u
        adicionar(cola, (v, distancia[v]))
```

### Tiempos de ejecución del algoritmo

Tiempo resultado de ejecutar el programa para distintos tamaños, empezando por un grafo de 64 vértices para consecutivamente ir duplicando su tamaño:



### Eficiencia de los algoritmos analizados teóricamente

- Algoritmo Dijkstra ( $\Theta(n^2)$ ):

El Algoritmo de Dijkstra realiza  $\Theta(n^2)$  operaciones (sumas y comparaciones) para determinar la longitud del camino más corto entre dos vértices de un grafo ponderado simple, conexo y no dirigido con  $n$  vértices.

- *Algoritmo mejorLocalizacion* ( $\Theta(n^4)$ ):

Se trata de un algoritmo iterativo formado por dos bucles, en el primer bucle realiza una llamada al algoritmo Dijkstra  $\Theta(n^2)$  y en el segundo realiza operaciones aritmético-lógicas. Por lo tanto su orden es de  $t(n) \in \Theta(n * n^2 * n) \rightarrow \Theta(n^4)$