MEMÒRIA

PRÀCTICA 1 COMPLEXITAT (DIJKSTRA)

<u>Dijkstra 1</u>

Per fer les proves per determinar les complexitats d'aquests algorismes, vam fer una funció que generés molts grafs aleatoris amb diferents nombres d'arestes i vèrtex, i calculés el temps d'execució dels dos algorismes. Per reduir al mínim l'error en el càlcul del temps d'execució, vam decidir mesurar el temps en executar cada algorisme 5 vegades per un mateix graf i després fer la mitjana.

Amb la funció Dijkstra 1, el primer que vam fer és fixar el nombre d'arestes a 1000, i vam anar augmentant el nombre de vèrtex, començant a 50 fins a arribar a 7550. Com es pot veure a la Figura 1, el temps va augmentant de forma quadràtica a mesura que augmentem el nombre de vèrtex.

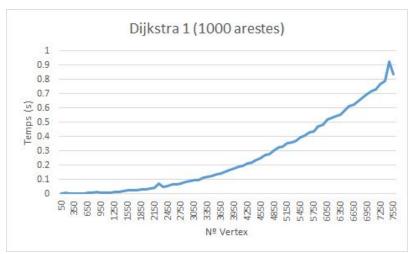


Figura 1: Algorisme de Dijkstra 1 guan augmentem el nombre de vèrtex (fixant el nombre d'arestes a 1000).

Un cop vist això, aquest cop vam fixar el nombre de vèrtex a 500 i vam anar augmentat el nombre d'arestes, però com es pot veure en la Figura 2, la complexitat d'aquest algorisme no depèn del nombre d'arestes, ja que el temps d'execució no es veu afectat quan les augmentem.

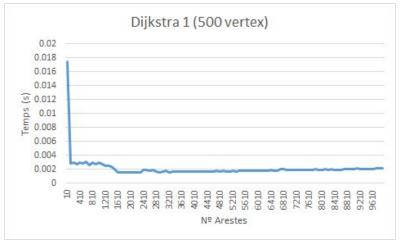


Figura 2: Algorisme de Dijkstra 1 quan augmentem el nombre d'arestes (fixant el nombre de vèrtex a 500).

Finalment, vam fer una ultima prova augmentant a la vegada tant el nombre d'arestes com de vèrtex. El resultat, com es pot veure a la Figura 3, es gairebé idèntic al de la Figura 1. Per tant, podem deduir que la complexitat d'aquest algorisme és quadràtica, depenent tan sols del nombre de vèrtex ($O(V^2)$)



Figura 3: Algorisme de Dijkstra 1 quan augmentem a la vegada vèrtex i arestes.

Dijkstra 2

Tal i com vam fer a l'algorisme Dijkstra 1, vam fixar el nombre d'arestes a 1000 i vam anar augmentant el nombre de vèrtex des de 50 a 7550. Com es pot veure a la Figura 4, el temps d'execució augmenta de forma lineal a mesura que augmentem els vèrtex.

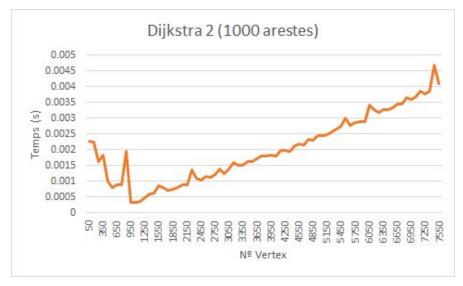


Figura 4: Algorisme de Dijkstra 2 quan augmentem el nombre de vèrtex (fixant el nombre d'arestes a 1000)

Seguidament, a l'igual que a l'algorisme 1, vam fixar el nombre de vèrtex a 500 i vam anar augmentant el nombre d'arestes des de 10 fins a 10000. En aquest cas però, el temps sí que canvia en funció del nombre d'arestes que hi posem (Figura 5).

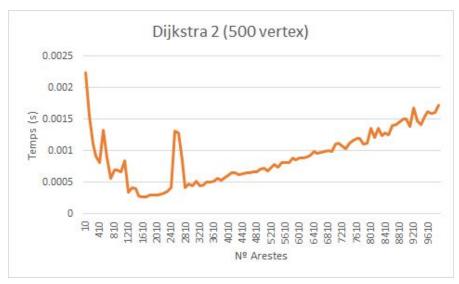


Figura 5: Algoritme Dijkstra 2 quan augmentem el nombre d'arestes (fixant el nombre de vertex a 500).

Per acabar, vam augmentar a la vegada els vèrtex i les arestes. El resultat, mostrat a la Figura 6, és molt semblant a les dues gràfiques anteriors. Amb això, podem deduir que la complexitat de l'algorisme Dijkstra 2 és gairebé lineal.



Figura 6: Algorisme de Dijkstra 1 quan augmentem a la vegada vèrtex i arestes.

Un cop vists per separat els dos algorismes, vam fer una gràfica comparant-los al canviar tant vèrtex com arestes, i amb els resultats obtinguts (Figura 7) veiem com clarament és molt més òptim l'algorisme de Dijkstra 2, ja que al utilitzar la cua de prioritat s'estalvia recórrer tots els nodes del graf a cada iteració del bucle. I això fa que la diferència de temps es vegi cada cop més clara a mesura que s'augmenta el nombre de vèrtex.

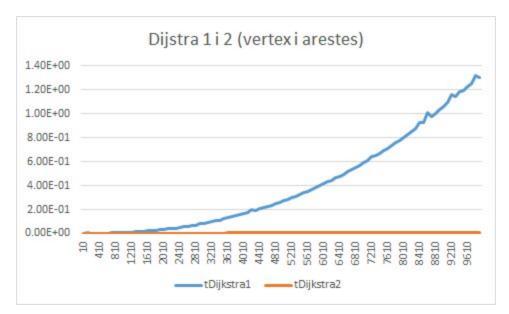


Figura 7: Comparació dels algorismes de Dijkstra 1 i 2 quan augmentem a la vegada vèrtex i arestes.

Per tant, al final hem comprovat que les complexitats que ens donava segons els dos algorismes són:

- Dijkstra 1: Complexitat quadràtica O(V²)
- Dijkstra 2: Complexitat lineal logarítmica O(A log(V))

El resultat, doncs, concorda amb el què ens havia de donar teòricament.