🚦 Mejorando el tráfico

por Daniela Leilany del Carmen Meléndez Rangel (102/20), Thiago Tiracchia (1502/21)

A dos estudiantes de algo3 les pidieron resolver el siguiente ejercicio:

Dado un mapa de una ciudad con n puntos, y m calles unidireccionales que conectan pares de puntos. Se ha propuesto una lista de k calles bidireccionales como candidatas a ser construidas con el fin de reducir la longitud entre los dos puntos críticos s y t.

Se debe escribir un programa para elegir la calle bidireccional de la lista propuesta tal que minimice la longitud resultante del camino más corto entre s y t.

Algoritmo

1. G <- inicializo grafo $G = (V, E, w)$ tq $V = \{\texttt{los n puntos en la ciudad}\}$, $E = \{\texttt{Las m calles unidireccionales}\}$ y $w(u, v)$ da el peso de ir de u a v que son puntos de la ciudad
2. Gt <- inicializo grafo $Gt = (V, Et, wt)$ tq $V = \{\texttt{los n puntos en la ciudad}\}$, $Et = (v, u) : (u, v) \in E$ y $wt(v, u) = w(u, v)$
3. d <- Dijkstra(G, s) //guardo en d las distancias desde s a cada nodo en G
4. dt <- Dijkstra(Gt, t) //guardo en dt las distancias desde t a cada nodo en Gt
5. nuevasCalles <- {}
6. para todo $(u, v, costo) \in$ {las k calles bidireccionales con el costo de ir de u hacia v ó viceversa}:
    a. nuevasCalles.push($(u, v, costo)$)
    b. nuevasCalles.push($(v, u, costo)$)
7. res <- calleMinimizadora(nuevasCalles, d, dT, t)

Donde `calleMinimizadora` es la función que busca la calle $(u, v) \in \texttt{nuevasCalles}$ tal que minimiza $d[u] + w(u, v) + d^t[v]$ y devuelve el mínimo entre $d[t]$ y $d[u] + w(u, v) + d^t[v]$.

Justificación

Sea (u, v) una propuesta de calle bidireccional, se cumplirá que $(u, v) \in$ **`nuevasCalles`** y que $(v, u) \in$ **`nuevasCalles`**, sea $d[k]$ la distancia que me da Dijkstra en el grafo G desde s hacia k, $\forall k \in V(G)$ y sea $d^t[k]$ la distancia que me da Dijkstra en el grafo G^t desde t hacia k, $\forall k \in V(G^t)$.

Como buscamos minimizar la longitud resultante del camino más corto entre s y t, para ello buscamos el mínimo entre $d[t]$, el camino mínimo original que tenía en el mapa original; y el $(u, v) \in$ **`nuevasCalles`** tal que minimice $d[u] + w(u, v) + d^t[v]$ entre todas las **`nuevasCalles`**.

Como sabemos que el mínimo debe estar entre alguno de **estos** dos valores?

Justificación para $d[t]$: Si no construimos ninguna calle, o todas las calles propuestas aumentan la longitud, sabemos que Dijkstra nos da el camino mínimo de s a todos los vértices de G, por lo que lo mínimo que puede valer es $d[t]$.

Justificación para $d[u] + d^t[v] + w(u, v)$: Si existe un camino mínimo de s a t que usa una calle propuesta $e = (u, v) \in$ **`nuevasCalles`**, sabemos que ese camino tendrá la forma $C = s, p_1, ..., p_n, u, v, q_1, ..., q_m, t$. Si cumple que es camino mínimo, por subestructura óptima sé que para todo par de nodos (a, b) en C tal que a sea menor en orden dado por C que b, $C_{a,b}$ es un camino mínimo de a a b. Por lo que

$$\begin{aligned} & C &=& \; C_{s,u} + C_{u,v} + C_{v,t} \\ \Rightarrow \; & d[C] &=& \; d[u] + d[u, v] + d^t[v] \qquad (1) \end{aligned}$$

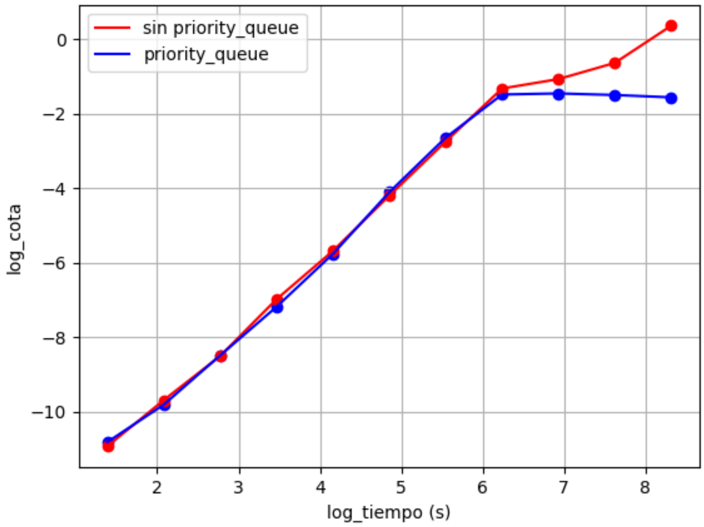
Entonces es correcto que la calle e que minimice a (1), será candidata a ser mínimo.

Experimentación

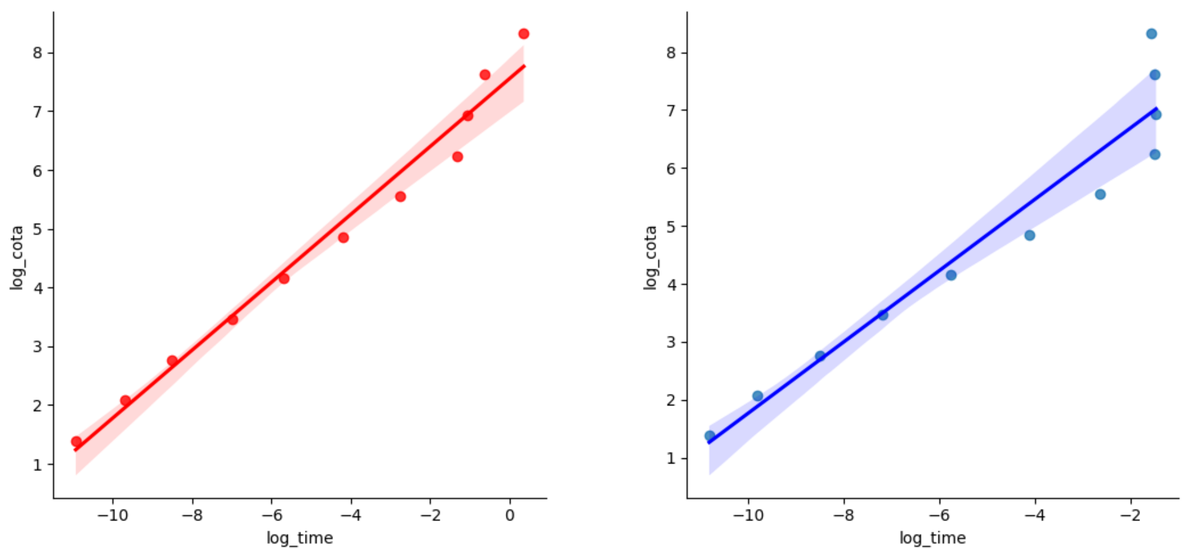
Usamos 2 implementaciones distintas de Dijkstra la cual difiren en que una está implementada con una cola de prioridad (azul) y en la otra se extrae el mínimo a procesar en $O(N)$ ya que lo busca por el vector de distancias linealmente (rojo). La complejidad teórica de la implementación con cola de prioridad es $O(V \cdot Log(V))$ si es un grafo ralo y $O(V^2)$ si es denso; en comparación a la complejidad teórica de la otra implementación que es $O(V^2)$ para cualquier grafo. En este ejercicio como $n \leq 10^4$ y $m \leq 10^5$ por ende, teóricamente va a depender del n cuál va a ser la mejor implementación porque mientras mas crezca el n, como el m esta acotado, menos denso va a hacer el grafo ya que no va a poder ser completo a partir de n = 10^3 ya que un grafo completo tiene $\frac{n \cdot (n-1)}{2}$ aristas.

Podemos ver que la experimentacion que con grafos densos son bastantes parecidas las dos implementaciones hasta un poco antes del final pero esto es debido a que m esta acotado y las aristas se acotan por 10^5 cuando el n crece mucho. Y que en la experimentacion de grafos ralos si puede notar la diferencia teórica esperada.

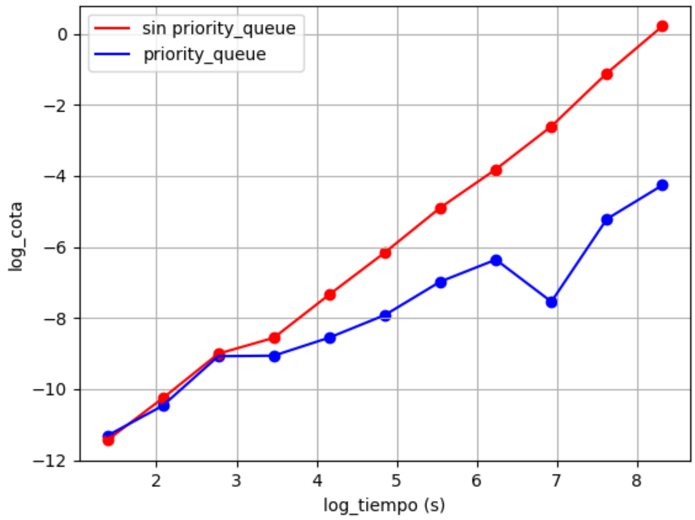
Experimentacion con Grafos Densos:



Comparacion con Grafos Densos



Experimentacion con Grafos Ralos:



Comparacion con Grafos Ralos