

HPPS

Informe 4 - Algoritmo de Dijkstra

Juan Braun

4 de junio de 2013

Introducción

El objetivo de este informe es comentar los resultados obtenidos para las diferentes versiones del algoritmo de Dijkstra implementadas. El algoritmo de Dijkstra es un algoritmo de búsqueda en grafos. Dado un grafo $G = (E, V)$ donde E son aristas con costos asociados y V los vértices, el algoritmo calcula los caminos de menor costo desde un vértice inicial $s \in V$ hasta todos los demás vértices $v \in V - \{s\}$.

A continuación se muestra el pseudo-código del algoritmo para el caso en que se usa la cola de prioridad.

```
dist[s] = 0; (distancia al vertice inicial)
for (todos los demas vertices)
    dist[v] = inf; (Las distancias a los nodos desconocidos es infinita)

S vacio; (En S voy guardando los nodos visitados)
build_heap(Q,V); (creo un min-heap Q con los vertices, se ordenan según el costo)

while (Q no vacio)'
    u = extractMin(Q) (saco el siguiente nodo con costo mas bajo)
    Agrego u a la lista de visitados S'

    for (todos los vecinos de u)

        if dist[v]>dist[u]+costo(u,v) (si hay un nuevo camino de menor costo)
            d[v] = d[u]+costo(u,v) (actualizo costo)
            move_up(Q,v) (Reordeno el heap)
        end if

    end for

return dist
```

El caso en el que no se usa la cola de prioridad es muy similar, la diferencia esta en que cada vez que se quiere conseguir el siguiente de menor costo se ordena el arreglo con N recorridas.

El pseudo-código que se muestra sirve para calcular los menores costos de un nodo inicial hacía todos los demás, lo que se necesita en este caso es encontrar el camino de menor costo entre dos nodos dados.

Para calcular el camino hacia un nodo en particular se necesita tener un arreglo en el que se almacenan los nodos previos con el costo mínimo para llegar a ellos. Con este arreglo y sabiendo que cualquier tramo incluido en un camino de costo mínimo es un camino de costo mínimo para un nodo en particular, es posible calcular el camino mínimo.

Por ejemplo para el grafo de la Figura 1, cuando se busca el camino entre el vértice A y el vértice F, se obtiene los siguientes arreglo con costos mínimos y nodos previos.

V	A	B	C	D	E	F	G	Z
Costos	0	25	20	25	25	20	10	30
Previo	nil	D	A	G	D	G	A	G

Cuadro 1: Costos y nodos previos

A partir del Cuadro 1 se puede calcular el menor camino de A a F. Se mira el nodo previo de F, G. Ahora se mira el nodo previo de G, A. El camino de menor costo es $A \rightarrow G \rightarrow F$

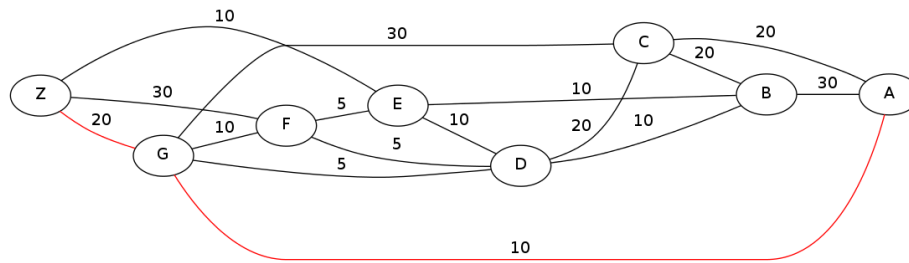


Figura 1: Grafo de prueba

La

1. Resultados

Se midieron los ciclos de reloj para diferentes entradas. Los resultados obtenidos se muestran el Cuadro 3.

	# nodos	# ciclos con heap	# ciclos sin heap
E1	8	21 260	198 015
E2	50	154 892	2 917 770
E3	75	279 164	7 780 416
E4	100	538 346	17 863 372
E5	250	2 718 002	204 572 291
E6	500	9 950 835	1 442 783 202
E7	750	21 601 507	4 652 164 219
E8	1000	37 042 054	10 718 859 780

Cuadro 2: Resultados

	# nodos	# ciclos con heap	# ciclos teoricos heap
E1	8	21 260	31
E2	50	154 892	331.2
E3	75	279 164	541.16
E4	100	538 346	763.4
E5	250	2 718 002	2240.4
E6	500	9 950 835	4981.4
E7	750	21 601 507	7912
E8	1000	37 042 054	10964

Cuadro 3: Resultados teo

	# relacion teorica	# relacion obtenida
E3/E2	1.64	1.81
E4/E3	1.4	1.92
E5/E4	2.94	5.04
E6/E5	2.22	3.66
E7/E6	1.59	2.17
E8/E7	1.39	1.71

Cuadro 4: Comparación resultados implementación con heap