

Programación Dinámica: Algoritmo de Floyd para caminos mínimos

Pamela Jiménez Rebenaque
Jesús Eduardo Plasencia Pimentel
Alberto Antonio Sarabia Suarez

Índice

1. Descripción del problema
2. Dijkstra
3. Floyd-Warshall
4. Evaluación experimental
5. Conclusiones
6. Bibliografía

Problema del camino mínimo

- Dado un grafo $G = (V, E)$
- Dado un par de nodos en E .
- Se busca el camino con la suma de los valores de las aristas mínima entre estos dos.

Dijkstra

Algoritmo voraz.

Consigue el camino mínimo entre sólo un par de vértices dado.

No funciona para grafos con costes negativos.

Pseudocódigo de Dijkstra

```
DIJKSTRA (Grafo  $G$ , nodo_fuente  $s$ )  
  para  $u \in V[G]$  hacer  
    distancia[ $u$ ] = INFINITO  
    padre[ $u$ ] = NULL  
  distancia[ $s$ ] = 0  
  Encolar (cola, grafo)  
  mientras que cola no es vacía hacer  
     $u$  = extraer_minimo(cola)  
    para  $v \in \text{adyacencia}[u]$  hacer  
      si distancia[ $v$ ] > distancia[ $u$ ] + peso ( $u$ ,  $v$ ) hacer  
        distancia[ $v$ ] = distancia[ $u$ ] + peso ( $u$ ,  $v$ )  
        padre[ $v$ ] =  $u$   
        Actualizar(cola,distancia, $v$ )
```

Análisis de complejidad

- Para cada nodo en E se debe revisar todas sus aristas conectadas, que será como máximo $|V|-1$. $O(|E|+|V|)$
- Conseguir y actualizar el peso de cada vértice es $O(\log|V|)$.
- Por tanto el tiempo total queda $O((|E|+|V|)\log|V|) = O(|E|\log|V|)$
- Para compararlo con Floyd hay que repetir esto con cada nodo como nodo inicial por tanto queda $O(|E|^2\log|V|)$.

Floyd-Warshall

Algoritmo de programación dinámica.

Consigue el camino mínimo entre todos los pares de vértices del grafo.

No funciona con grafos con ciclos de valores negativos.

Pseudocódigo de Floyd

```
let dist be a  $|V| \times |V|$  array of minimum distances initialized to  $\infty$  (infinity)
for each vertex  $v$ 
     $\text{dist}[v][v] \leftarrow 0$ 
for each edge  $(u,v)$ 
     $\text{dist}[u][v] \leftarrow w(u,v)$  // the weight of the edge  $(u,v)$ 
for  $k$  from 1 to  $|V|$ 
    for  $i$  from 1 to  $|V|$ 
        for  $j$  from 1 to  $|V|$ 
            if  $\text{dist}[i][j] > \text{dist}[i][k] + \text{dist}[k][j]$ 
                 $\text{dist}[i][j] \leftarrow \text{dist}[i][k] + \text{dist}[k][j]$ 
            end if
```


Análisis de complejidad de Floyd

- Siendo $n = |V|$
- Se busca el camino mínimo entre cada par de nodos i, j en V . $O(n^2)$
- Se va mejorando el valor a lo largo de n iteraciones. $O(n)$
- Al combinar ambas operaciones resulta $O(n^3)$

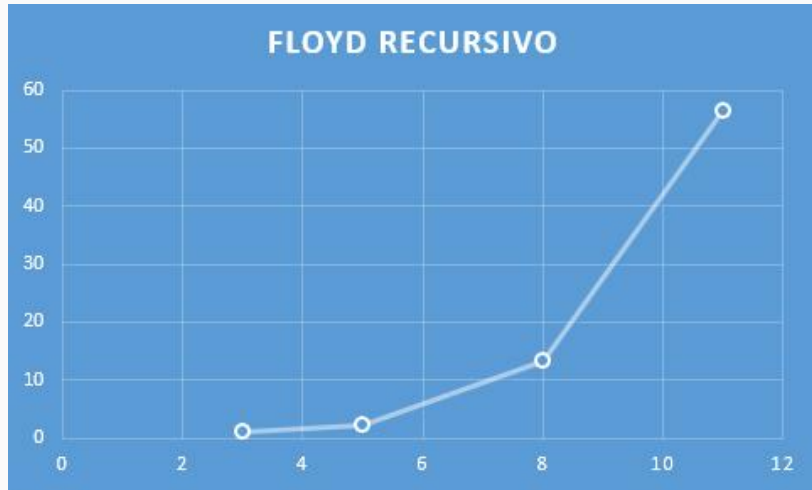
Evaluación experimental

Número de nodos (n)	Dijkstra (ms)	Floyd recursivo (ms)	Floyd iterativo (ms)
3	1,283660	1,105734	0,926862
5	2,780796	2,233947	1,622145
8	3,853728	13,326303	2,145761
11	11,350863	56,636881	2,558855

Evaluación experimental



Evaluación experimental



Conclusiones

Para los casos probados Floyd iterativo supera a Dijkstra.

Sin embargo haciendo el análisis de complejidad podemos ver que Dijkstra puede superar a Floyd en ciertos grafos y con la implementación correcta.

Por último, Dijkstra no es una opción válida para resolver caminos mínimos en un grafo con costes negativos.

Gracias por su atención.

Bibliografía

https://es.wikipedia.org/wiki/Problema_del_camino_m%C3%A1s_corto

https://en.wikipedia.org/wiki/Floyd%E2%80%93Warshall_algorithm#Algorithm

https://es.wikipedia.org/wiki/Algoritmo_de_Dijkstra

https://www.ecured.cu/Algoritmo_de_Dijkstra

<http://algoritmodefloyd.blogspot.com.es/>