



Practica No.6

Calculo del camino más corto entre los nodos de un grafo

(Algoritmo de Dijkstra)

Nombre(s):

Santiago André Vital Betanzos

Objetivo:

Con la realización de esta práctica se pretende que: se calcule la distancia más corta entre los nodos de un grafo en ANSI C.

Fundamento Teórico:

Algoritmo de Dijkstra

Uno de los algoritmos más usados para la búsqueda de caminos de peso mínimo es el de Dijkstra, que proporciona los pesos mínimos desde un nodo dado al resto de los nodos.

El algoritmo devuelve en realidad el peso mínimo no el camino mínimo propiamente dicho, pero permite obtener fácilmente el camino mínimo recorriendo en sentido inverso la construcción (de ahí su popularidad).

Sea un grafo pesado, con $V=\{v_1, v_2, \dots, v_n\}$ su conjunto de nodos y su matriz de pesos $\Omega = \omega_{ij}^{n \times n}$, y sea v_p el nodo inicial.

Dijkstra construye, en cada paso, un camino mínimo desde v_p a otro nodo y se detiene cuando ha construido uno para cada nodo (o no puede construir más). Para ello se usan una lista o conjunto: L , que contendrá los nodos para los que ya hemos construido un camino mínimo y un vector de pesos:

D , que contendrá al final los pesos mínimos. Inicialmente $L=\{v_p\}$ y $D=\Omega(p, :)$, la p -ésima fila de la matriz de pesos (la correspondiente al nodo inicial).

Algoritmo 4.- (de Dijkstra)

```
inicio:  $\Omega; v_p; L = \{v_p\}; D = \Omega(p, :)$ 
mientras sea  $V - L \neq \emptyset$ 
    tomar  $v_k \in V - L$  con  $D(k)$  mínimo
    hacer  $L = L \cup \{v_k\}$ 
    para cada  $v_j$  de  $V - L$ 
        si  $D(j) > D(k) + \omega_{kj}$ 
            hacer  $D(j) = D(k) + \omega_{kj}$ 
    fin
fin
```

El vector D final contiene los pesos mínimos desde el nodo inicial a los demás nodos -si alguno de los pesos finales es ∞ , no hay camino desde el nodo inicial-.

El algoritmo de Dijkstra es un algoritmo voraz (*greedy*) que genera uno a uno los caminos de un nodo a al resto por orden creciente de longitud; usa un conjunto S de vértices donde, a cada paso del algoritmo, se guardan los nodos para los que ya se sabe el camino mínimo y devuelve un vector indexado por nodos, de modo que para cada uno de estos nodos se puede determinar el costo de un camino más económico (de peso mínimo) de a a tales nodos.

Cada vez que se incorpora un nodo a la solución, se comprueba si los caminos todavía no definitivos se pueden acortar pasando por él.



Forma de trabajo:

Colaborativa en equipos de 2 personas

Material:

1. Computadora
2. Compilador ANSI C

Procedimiento:

Se va a crear un programa que calcule la distancia más corta entre el vértice A y H mediante el algoritmo de Dijkstra del siguiente grafo:

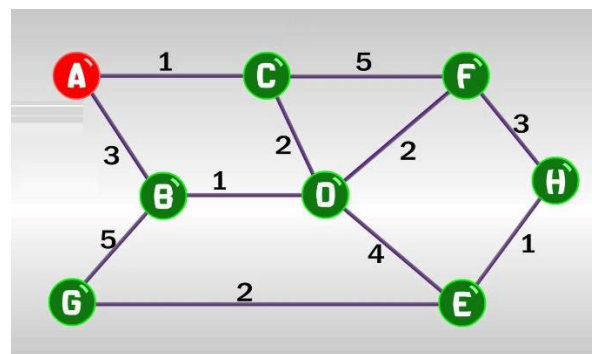


Figura 2 Grafo conexo

Para la creación del programa deberán realizarse los siguientes pasos:

1. En las primeras líneas elaborar comentarios con la siguiente información:
 - a. Nombre de la institución
 - b. Nombre de la carrera
 - c. Nombre de la materia
 - d. Nombre(s) de quien(es) realiza(n) la práctica
 - e. Nombre del profesor
 - f. Una descripción breve de lo que realiza el programa
2. Definir la matriz de pesos del grafo pesado.
3. Desplegar en pantalla la matriz de pesos para mostrar el grafo.
4. Crear un menú que contenga las siguientes opciones:
 - a. Calcular la distancia más corta entre el vértice A y el H del grafo y desplegar el camino resultante y la distancia recorrida.
 - b. Una opción para salir



Resultados:

Realizar al menos dos corridas de prueba y mostrar imágenes de las pantallas de texto generadas.





Conclusiones:

El algoritmo de Dijkstra es el procedimiento más eficiente que conocemos a día de hoy para dar solución al problema de todos los caminos más cortos de un grafo. También se puede observar que tiene sus limitaciones, tal como el hecho de que el grafo no puede contener pesos negativos.

Una vez terminado el programa fuente (.c) debe subirse a la plataforma de **aulavirtual** junto con este reporte.