**UNIVERSIDAD ABIERTA PARA ADULTOS**

**Asignatura:**

Estructura de Datos y Algoritmos

**Tema:**

Proyecto Final

**Participantes:**

Christian Medrano

Johan García Ramírez

Ryan Rene Sánchez

Dawlin Diaz Figueroa

Rigoberto Rodriguez

Carlos Cueva

Francis Gómez

**Matriculas:**

1000-29253

1000-16023

1000-31794

1000-20389

1000-16189

1000-38032

1000-24212

**Facilitador:**

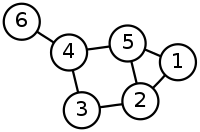
Maikel Aparicio

**Fecha:** 25/3/2022

* **Estructuras de datos grafos:**

Un grafo en el ámbito de las ciencias de la computación es un tipo abstracto de datos (TAD), que consiste en un conjunto de nodos (también llamados vértices) y un conjunto de arcos (aristas) que establecen relaciones entre los nodos. El concepto de grafo TAD desciende directamente del concepto matemático de grafo.

Formalmente, un grafo se define como {\displaystyle G=(V,A)}{\displaystyle G=(V,A)}, siendo V un conjunto cuyos elementos son los vértices del grafo y A un conjunto cuyos elementos son las aristas, las cuales son pares (ordenados si el grafo es dirigido) de elementos en V.



* **Algoritmo de Dijkstra:**

El algoritmo de Dijkstra, también llamado algoritmo de caminos mínimos, es un algoritmo para la determinación del camino más corto dado un vértice origen al resto de los vértices en un grafo con pesos en cada arista. Su nombre se refiere a *Edsger Dijkstra*, quien lo describió por primera vez en 1959.

La idea subyacente en este algoritmo consiste en ir explorando todos los caminos más cortos que parten del vértice origen y que llevan a todos los demás vértices; cuando se obtiene el camino más corto desde el vértice origen, al resto de vértices que componen el grafo, el algoritmo se detiene. El algoritmo es una especialización de la búsqueda de costo uniforme, y como tal, no funciona en grafos con aristas de coste negativo (al elegir siempre el nodo con distancia menor, pueden quedar excluidos de la búsqueda nodos que en próximas iteraciones bajarían el costo general del camino al pasar por una arista con costo negativo).

Este algoritmo utiliza dos conjuntos de nodos, SS y CC. SS contiene los nodos ya seleccionados y cuya distancia mínima al origen ya se conoce y CC contiene los demás nodos, C=N/SC=N/S, aquellos cuya distancia mínima al origen no se conoce todavía. Al inicio del algoritmo SS sólo contiene el nodo origen y cuando finaliza el algoritmo contiene todos los nodos del grafo y además se conocen las longitudes mínimas desde el origen a cada uno de ellos. La función de selección elegirá en cada paso el nodo de CC cuya distancia al origen sea mínima.

* **Ejemplo:**

**tipo** VectorNat = matriz[O..n] de natural

**fun** Dijkstra (G = : grafo): VectorNat, VectorNat

**var**

especial, predecesor: VectorNat

C: conjunto de nodos

**fvar**

C={2,3,...,n}

**para** i = 2 **hasta** n **hacer**

especial[i] = Distancia(1 ,i)

predecesor[i] = 1

**fpara**

**mientras** C contenga más de 1 nodo **hacer**

v = nodo en C que minimiza especial[v]

C = C{v}

**para** cada w en C **hacer**

**si** especial[w] > especial[v] + Distancia(v,w) entonces

especial[w] = especial[v] + Distancia(v,w)

predesecor[w] = v

**fsi**

**fpara**

**fmientras**

**dev** especial[],predecesor[]

**ffun**

