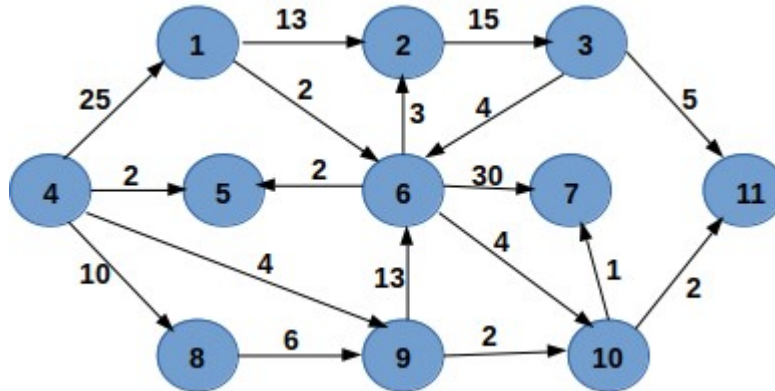


TAREA 2. Algoritmo de Dijkstra e interpretación

El algoritmo de Dijkstra permite encontrar los **caminos mínimos**, si existen, desde un vértice inicial al resto de los vértices de un grafo dirigido ponderado. En esta tarea hay que implementar las dos versiones (básica y mejorada) de este algoritmo, aplicárselo al grafo de la figura 1 y codificar las funciones que permiten interpretar el resultado del algoritmo.



- figura 1-

Las especificaciones que hay seguir son las siguientes:

1. Representación de grafos mediante listas de adyacencia como se indica en teoría (T4.Grafos.pdf) y prácticas (T4.Prácticas Grafos.pdf).
2. Implementar las dos versiones del algoritmo de Dijkstra según los siguientes prototipos:

- versión básica: **void dijkstra1(int vInicio, tipoGrafo *g)**

- versión mejorada Montículo Binario: **void dijkstra2(int vInicio, tipoGrafo *g)**

3. Implementar las funciones que permiten interpretar los resultados del algoritmo de Dijkstra, obtención de caminos y distancia entre vértices:
 - función que muestra la trayectoria (secuencia de vértices) del camino mínimo, si existe, entre un vértice inicial (parámetro vIni) y un vértice final (parámetro vFin), después de haber aplicado cualquiera de los algoritmos del punto 2 sobre un grafo (tercer parámetro). Esta función devuelve un entero que indica la distancia de ese camino mínimo.

int costeYTrayectoria(int vIni, int vFin, tipoGrafo *g)

- función que muestra todas las trayectorias de caminos mínimos y sus distancias, si existen, desde un vértice inicial (parámetro vIni) al resto de los vértices, después de haber aplicado cualquiera de los algoritmos del punto 2, de un grafo (segundo parámetro).

void todosCaminosMin(int vIni, tipoGrafo *g)

4. Implementar una función que cree y devuelva el grafo de la figura 1 y observar sobre él la correcta ejecución de las funciones implementadas en los apartados anteriores.

tipoGrafo *creaGrafo()

TAREA 2. Algoritmo de Dijkstra e interpretación

Se adjuntan los siguientes ficheros de ayuda en tarea2.zip :

prueba.c fichero de prueba del algoritmo de Dijkstra y de algunos más. Se valorará la inclusión de las pruebas de otros algoritmos que no se piden pero que deberíais tener implementados.

makefile para la gestión de este pequeño proyecto.

tarea2.Dijkstra.pdf con el enunciado de la práctica

Notas:

- No se adjuntan los ficheros **grafos.h** y **grafos.c** porque son los mismos que tenéis en la práctica de grafos, con los que lleváis trabajando dos semanas.
- En grafos.c se incluye implementada la función verGrafo(g) que es obligatorio utilizar en las pruebas.
- El fichero prueba.c es una modificación del ejercicio3.c que tenéis en los ficheros de ayuda para la realización de las prácticas de grafos.

Condiciones de Entrega. Debéis entregar un fichero comprimido que contenga:

- **grafos.c** con la implementación de las funciones solicitadas en las especificaciones y todas aquellas, referentes a grafos, que estos algoritmos invoquen (por ejemplo iniciar(g)).
- **imagen** o **fichero.txt** con el resultado de ejecución de prueba.c
- **prueba2.c** fichero prueba.c modificado para que los algoritmos se ejecuten sobre el grafo de la figura 1 (punto 4 de las especificaciones). **La ejecución de este código debe obtener los caminos mínimos desde el vértice 2 y desde el vértice 10 al resto de los vértices del grafo de la figura 1.**
- **imagen2** o **fichero2.txt** con el resultado de ejecución de prueba2.c

Hay que entregar solamente los **cuatro** ficheros que se piden. Si no están los resultados de ejecución (imagen o fichero de texto) o si la situación inicial que muestren estos resultados no es correcta, se considerará entrega incorrecta.

