

Desarrollar un programa en C++ que modele un grafo con al menos 15 vértices, máximo 30, donde el usuario pueda capturar dos vértices y el programa deberá encontrar las 2 mejores rutas posibles entre ellos.

La cantidad de vértices debe ser elegida de forma aleatoria (15-30).

Cada arista entre vértices deberán ser generadas aleatoriamente con mínimo 2 relaciones, máximo=  $N/3$ . (Ejemplo  $20/3 \approx 7$  aristas). Ponderándolo aleatoriamente.

Restricciones:

1. El programa deberá representar un grafo dirigido ponderado.
2. El grafo debe contener al menos 15 vértices o nodos y puede estar representado mediante una matriz de adyacencia o listas de adyacencia.
3. El programa debe permitir al usuario capturar dos vértices (de inicio y de destino) mediante entrada por teclado.
4. Utilizar algoritmos de búsqueda como DFS (Depth-First Search) o BFS (Breadth-First Search) para encontrar las 2 mejores rutas entre los vértices capturados.
5. Si no existe una ruta entre los vértices, el programa deberá indicarlo.
6. Antes de realizar la captura de los vértices de inicio y destino, el programa debe mostrar la matriz de adyacencia o la lista de adyacencia