

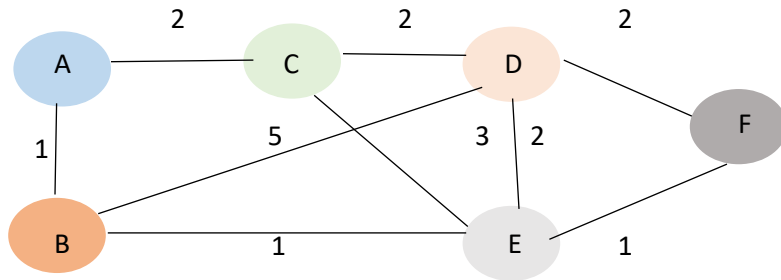
Laboratorio Grafos

David Leonardo León V.

Jhon Corredor

30/10/2024

- Primer punto: Grafo del código.



- Segundo punto: Entrada actualizada del código con el grafo de la clase:

```

int main() {

    int graph[7][7] = {
        {0, 2, 4, 0, 0, 0, 0},
        {2, 0, 0, 5, 0, 0, 0},
        {4, 0, 0, 8, 0, 0, 0},
        {0, 5, 8, 0, 10, 15, 0},
        {0, 0, 0, 10, 0, 6, 2},
        {0, 0, 0, 15, 6, 0, 6},
        {0, 0, 0, 0, 2, 2, 0}
    };

    DijkstraAlgo(graph, 0, 7);
    return 0;
}

```

- Documentación código:

Con este código podemos encontrar la distancia mínima desde un vértice fuente a todos los demás vértices en un grafo ponderado, representado mediante una matriz de adyacencia. La función `miniDist` encuentra el vértice con la menor distancia que aún no ha sido procesado. La función principal `DijkstraAlgo` inicializa las distancias desde el vértice fuente a todos los vértices como infinitas y luego, en cada iteración, selecciona el vértice con la menor distancia. Si se encuentra un

camino más corto hacia un vértice adyacente, la distancia se actualiza. Al final, se imprime la distancia mínima desde la fuente a cada vértice. En main, se define el grafo con una matriz de adyacencia basada en los datos y se llama a `ijkstraAlg` para calcular y mostrar las distancias.