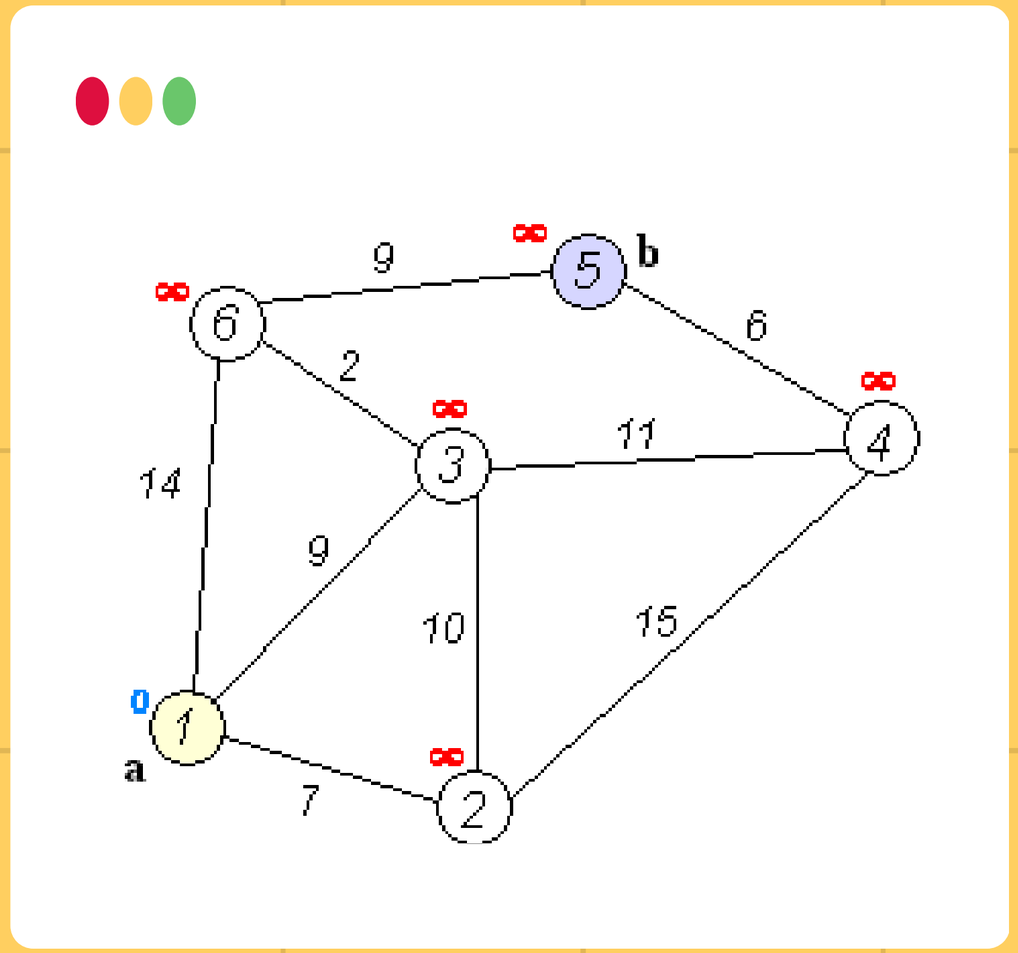
Dijkstra

Luis Rojas

Definición

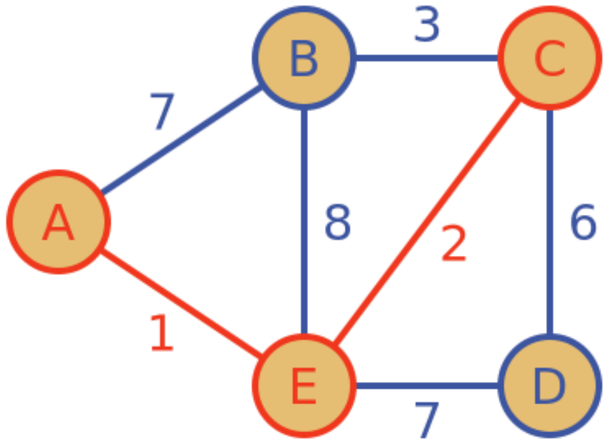
Es un algoritmo de búsqueda de caminos más cortos en un grafo ponderado con pesos no negativos.

Encuentra la distancia mínima desde un nodo origen hasta todos los demás nodos.



Idea principal

En cada paso, selecciona el nodo con la distancia más corta conocida y actualiza las distancias de sus vecinos.



	Vis	Dist	Prev
A	1	0	-
B	0	6	C
C	1	3	E
D	0	8	E
E	1	1	A

Pasos

- Inicializar todas las distancias en ∞, excepto la del nodo origen (0).
- Marcar todos los nodos como no visitados.
- Seleccionar el nodo con la distancia más corta no visitado.
- Actualizar las distancias de sus vecinos si se encuentra un camino más corto.
- Marcar el nodo como visitado.
- Repetir hasta visitar todos los nodos o llegar al destino.

O((V+E)log V)
V = nro Vertices
E = nro de aristas

```cpp
void dijkstra(int n, int start, vector<vector<Edge>>& adj) {
    vector<int> dist(n, INF);   // distancias mínimas
    dist[start] = 0;

    // Min-heap (distancia, nodo)
    priority_queue<pair<int,int>, vector<pair<int,int>>, greater<>> pq;
    pq.push({0, start});

    while (!pq.empty()) {
        auto [d, u] = pq.top(); pq.pop();

        if (d > dist[u]) continue; // ya se encontró un mejor camino

        for (auto edge : adj[u]) {
            int v = edge.to;
            int w = edge.weight;

            if (dist[u] + w < dist[v]) {
                dist[v] = dist[u] + w;
                pq.push({dist[v], v});
            }
        }
    }
}
```