

Graphs

A graph is a pair of sets: a set of vertices V and a set of edge E .

An edge will be

- a source vertex
- a destination vertex
- a weight
- a label

Kinds of graphs

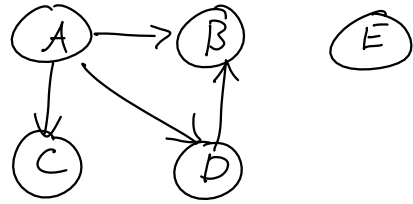
- undirected/directed
- unweighted/weighted
- disconnected/connected/weakly connected

Graph ADT

V - vertex type
 E - label type
 W - weight

```

void insertVertex(V vertex)
void insertEdge(V src, V dest, W weight, E label)
vector<V> getNeighbors(V vertex) ← only outgoing edges
vector<Edge<V,E,W>> getIncomingEdges(V vertex)
vector<Edge<V,E,W>> getOutgoingEdges(V vertex)
bool containsVertex(V vertex)
Edge<V,E,W> getEdge(V src, V dest)
void removeVertex(V vertex)
void removeEdge(V src, V dest)
void updateWeight(V src, V dest, W weight)
    
```

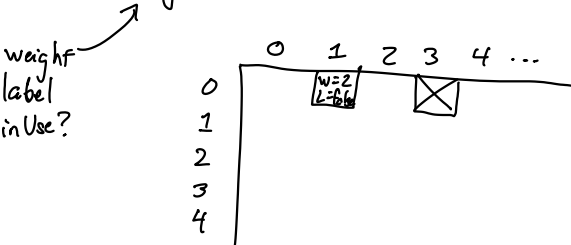
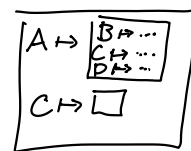
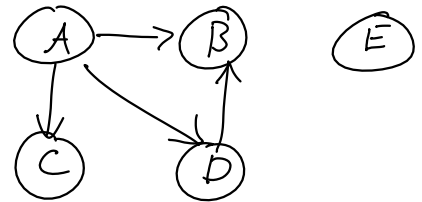


```

template <V, E, W>
class Edge {
    V src;
    V dest;
    E label;
    W weight;
};
    
```

Implementations

- ① Hash Table $\langle V, \text{Array List} \langle \text{Edge} \langle V, E, W \rangle \rangle \rangle$
- ② Hash Table $\langle V, \text{Hash Table} \langle V, \text{Edge} \langle V, E, W \rangle \rangle \rangle$
- ③ EdgeData [] []



- Ⓐ hash vertices
- Ⓑ Dictionary $\langle V, \text{int} \rangle$

Adjacency List Graph

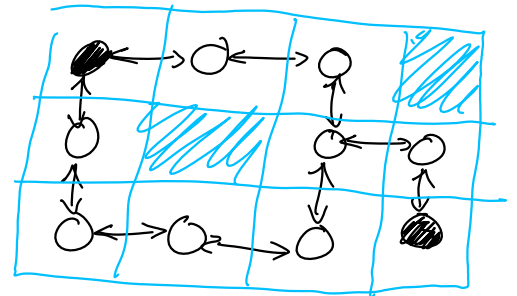
Adjacency Matrix Graph

- ④ Linear Graph

What to do with graphs?

- * shortest path from V_1 to V_2
- * is there a path that includes each edge exactly one time? (or each vertex once)
- * find all paths from V_1 to V_2
- * transform directed to undirected
- * is there any path from V_1 to V_2 ?

Function ^{BFS} reachable-DFS (Graph g , V_{start} , V_{end}):
exploration \leftarrow new Stack \leftarrow Queue
exploration \leftarrow insert(start)
visited \leftarrow new Set \leftarrow you can use a Dictionary $\langle V, \text{bool} \rangle$
visited.add(start)



While exploration is not empty:

current \leftarrow exploration.remove()

If current = end: Return true

For each neighbor in $g.getNeighbors(current)$:

If neighbor is not in visited:

exploration.insert(neighbor)

visited.add(neighbor)

End If

End For

End While

Return false

End Function

