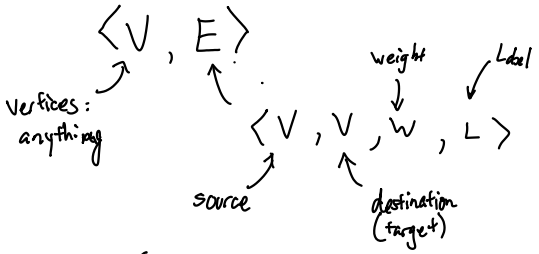
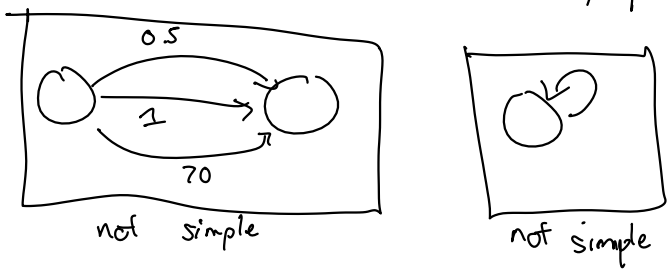
# Graphs

What is a graph?



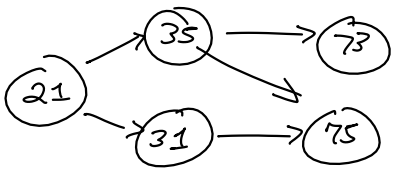

"Simple" graph: has no self-loops (no edges where source = destination).
only one edge between any pair of vertices




# ADT — Graph

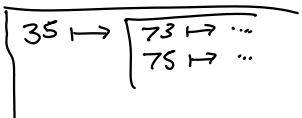
```
Edge<V,W,L> getEdge(V src, V dst)
void insertEdge(V src, V dst, W weight, L label)
vector<Edge<V,W,L>> getOutgoingEdges(V src)
vector<Edge<V,W,L>> getIncomingEdges(V dst)

vector<V> getVertices()
void insertVertex(V vertex)
bool containsVertex(V vertex)
void removeEdge(V src, V dst)
void removeVertex(V vertex)
vector<V> getNeighbors(V vertex)
bool containsEdge(V src, V dst)
```

```
template <typename V, typename W, typename L>
class Edge {
    V src;
    V dst;
    W weight;
    L label;
}
```

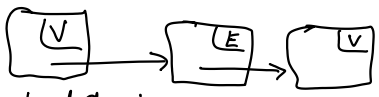



# Graph Implementations

1. `Dictionary<V, Dictionary<V, Edge<V,W,L>>>`
   - src → dst
   - AdjacencyListGraph / AdjacencyMapGraph



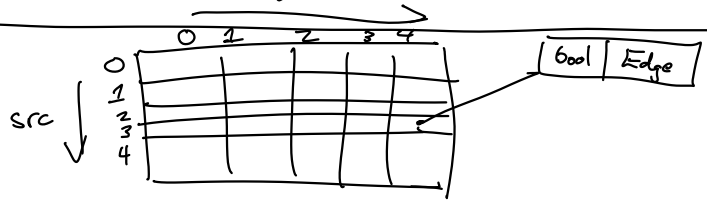

`Dictionary<V, Dictionary<V, Edge<V,W,L>>>`
dst, src

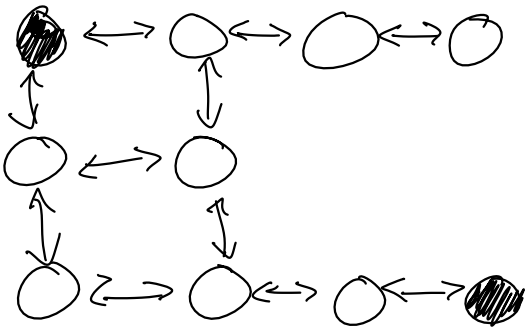
2. 

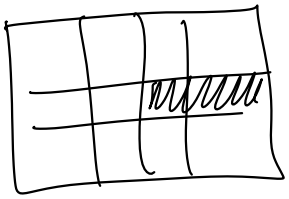
LinkedGraph

3. Adjacency Matrix Graph

`Dictionary<V, int>`







```
Function reachableDFS(graph, src, dst) : bool
    frontier ← new Stack<V>
    visited ← new Set<V>
    frontier.insert(src)
    visited.add(src)
    while frontier is not empty:
        v ← frontier.remove()
        If v == dst:
            Return true
        End If
        For each neighbor of v not in visited:
            frontier.insert(v)
            visited.add(v)
        End For
    End While
    Return false
```

Set<V>

=

Dictionary<V, ??>

int=0