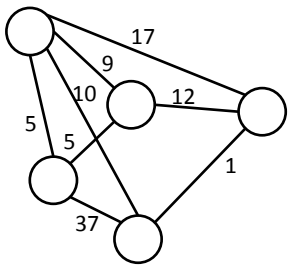
Weighted Graphs I

University of Massachusetts Amherst
ECE 242 – Data Structures and Algorithms
Lecture 26

Weighted graphs

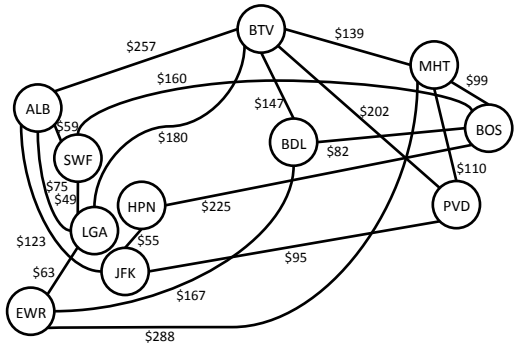
- Edges have "weights"
  - Cost metric
- Example interpretations
  - Distance
  - Dollar cost of traversal
  - Delay
- Cost is assumed to be additive



Weighted tree example

- Example: cost of flying between airports
  - "0" means no edge

	ALB	BDL	LGA	JFK	EWR	HPN	PVD	BOS	MHT	BTV	SWF
ALB	0	0	75	123	0	0	0	0	0	257	59
BDL	0	0	0	0	167	0	0	82	0	147	0
LGA	75	0	0	0	63	0	0	0	0	180	49
JFK	123	0	0	0	0	55	95	0	0	0	0
EWR	0	167	63	0	0	0	0	0	288	0	0
HPN	0	0	0	55	0	0	0	225	0	0	0
PVD	0	0	0	95	0	0	0	0	110	202	0
BOS	0	82	0	0	0	225	0	0	99	0	160
MHT	0	0	0	0	288	0	110	99	0	139	0
BTV	257	147	180	0	0	0	202	0	139	0	0
SWF	59	0	49	0	0	0	0	160	0	0	0

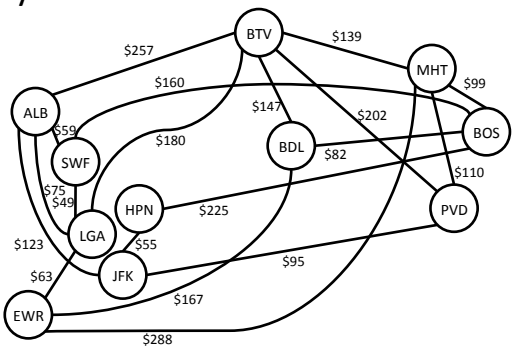


Algorithms for weighted graphs

- Weights make algorithms more interesting
- Example: minimum spanning tree
  - Find cheapest set of flights to connect all airports
- Problem
  - Why does our previous algorithm not work?
  - Can you provide example where it fails?

Minimum spanning tree

- How to find set of edges with minimum total weight to connect all nodes?
  - Not easy to "see" correct solution



Ideas

- Add edges that are cheap
  - If node still needs to be connected, might as well use a cheap edge
- Don't add edges between nodes that have already been connected
  - Redundant edge not necessary

Kruskal's algorithm

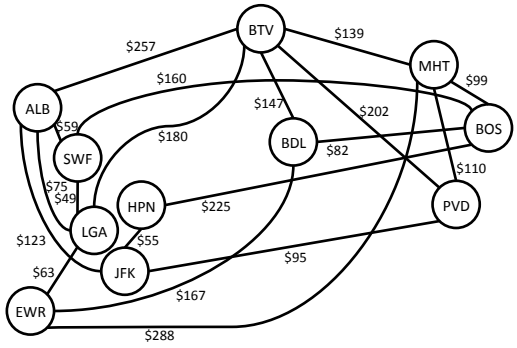
- Assign each vertex to its own set
- Add all edges to priority queue (descending)
- For each edge in priority queue
  - If vertices are in different sets
    - Add edge to MST
    - Merge sets

Prim's algorithm

- Alternative implementation of MST
  - Used in textbook
- Pick one vertex from graph and assign to set
- Repeat until set includes all nodes
  - Find least-cost edge connecting new vertex to set
    - Add edge
    - Add vertex to set

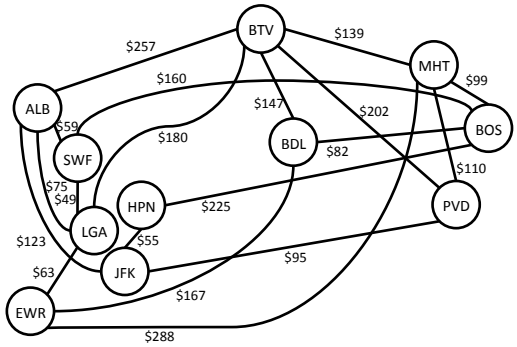
Kruskal's algorithm example

- Try Kruskal's algorithm yourself!



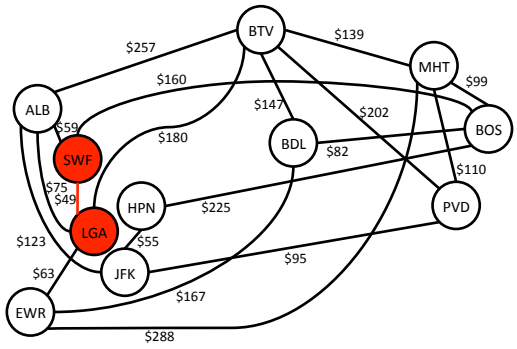
Kruskal's algorithm example

- Initial setup: all nodes in separate sets



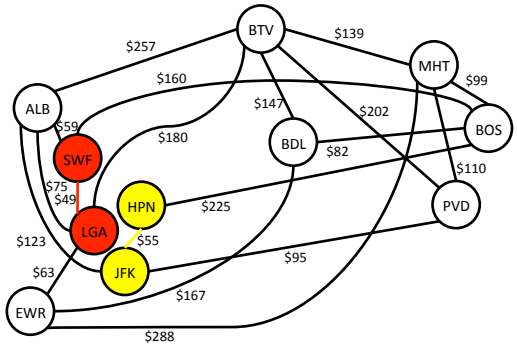
Kruskal's algorithm example

- Find least-cost edge: SWF-LGA $49



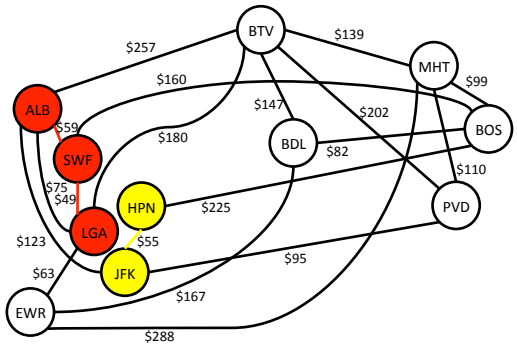
Kruskal's algorithm example

- Next least-cost edge: JFK-HPN $55



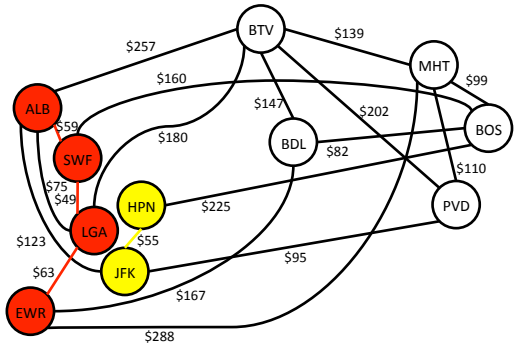
Kruskal's algorithm example

- Next least-cost edge: ALB-SWF $59



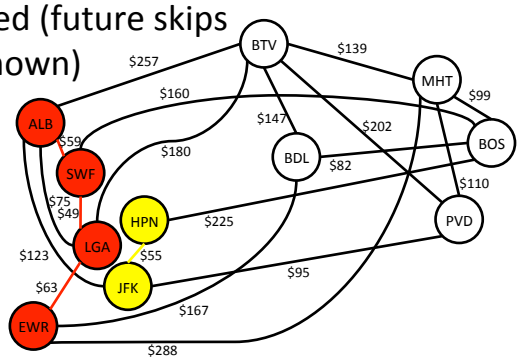
Kruskal's algorithm example

- Next least-cost edge: EWR-LGA $63



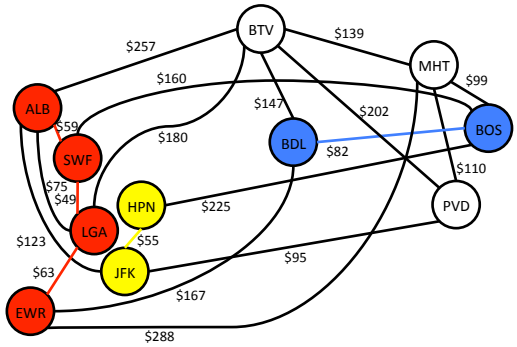
Kruskal's algorithm example

- Next least-cost edge: ALB-LGA $75
  - Skipped (future skips not shown)



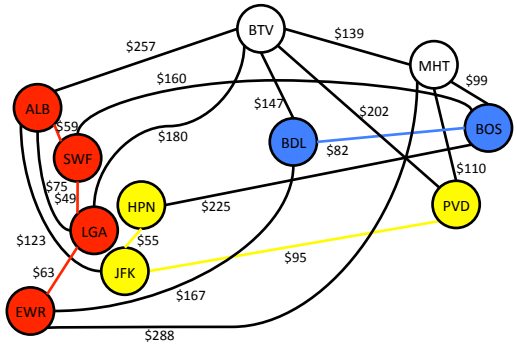
Kruskal's algorithm example

- Next least-cost edge: BOS-BDL $82



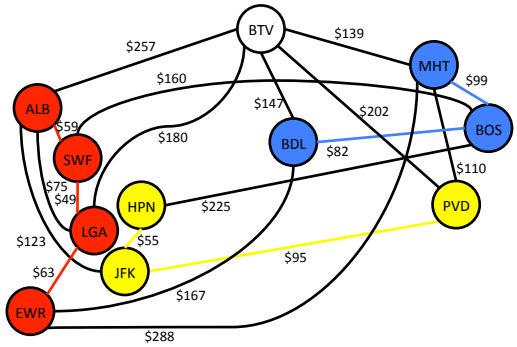
Kruskal's algorithm example

- Next least-cost edge: JFK-PVD $95



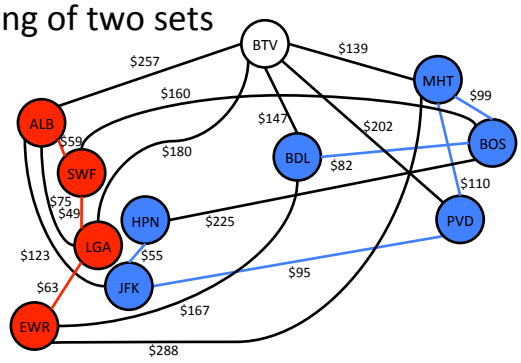
Kruskal's algorithm example

- Next least-cost edge: BOS-MHT $99



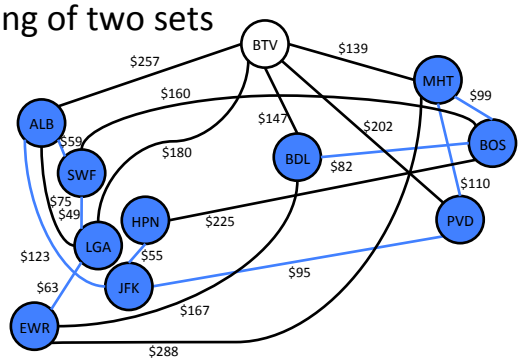
Kruskal's algorithm example

- Next least-cost edge: PVD-MHT $110
  - Merging of two sets



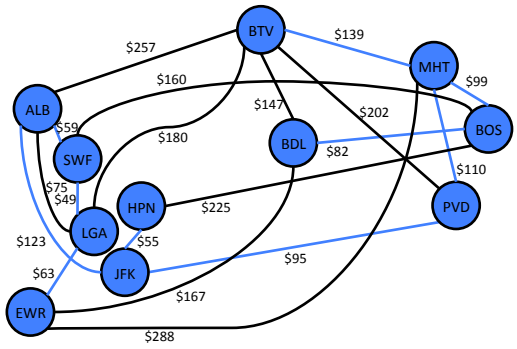
Kruskal's algorithm example

- Next least-cost edge: ALB-JFK $123
  - Merging of two sets



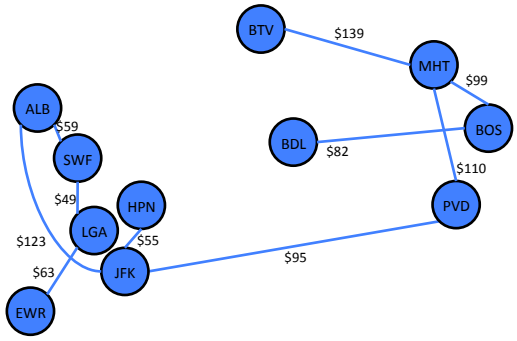
Kruskal's algorithm example

- Next least-cost edge: BTV-MHT $139



Kruskal's algorithm example

- Result:



Code

```
public Graph minimumSpanningTree() {
    Graph mst = new Graph(maxVertices); // create new graph
    int[] set = new int[activeVertices];
    for (int i=0; i<activeVertices; i++) { // copy nodes to graph
        mst.addVertex(vertices[i]);
        set[i]=i; // assign each node to its own set
    }
    PriorityQueue q = new PriorityQueue(maxVertices*maxVertices);
    // create priority queue
    for (int i=0; i<activeVertices; i++) { // copy edges to queue
        for (int j=0; j<activeVertices; j++) {
            if (edges[i][j]!=0) {
                q.enqueue(new Edge(vertices[i],
                vertices[j],edges[i][j]));
            }
        }
    }
    ...
```


Code

```
    ...
    while (!q.isEmpty()) { // iterate over all edges in priority order
        Edge e = q.dequeue(); // consider next edge
        if (set[e.source.graphIndex]!=set[e.destination.graphIndex]) {
        // skip edges not connecting different sets
            mst.addEdge(e.source, e.destination, e.weight);
            // add edge to MST
            System.out.println("adding "+e);
            int setToMerge=set[e.destination.graphIndex];
            // rename nodes from "other" set
            for (int i=0; i<activeVertices; i++) {
                if (set[i]==setToMerge) { // find nodes from "other" set
                    set[i]=set[e.source.graphIndex]; // reassign nodes
                }
            }
        }
    }
    return mst;
}
```

Next Steps

- HW due on Thursday
- No class on Friday