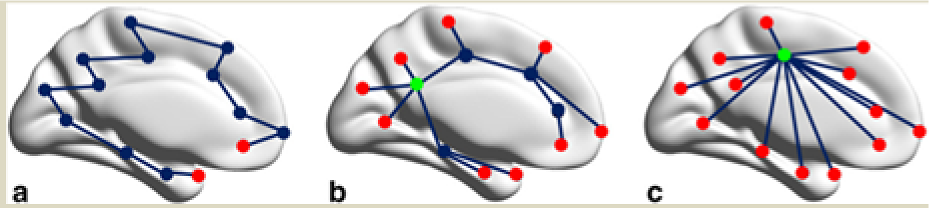
Lecture14



CSE 417
Algorithms and Complexity

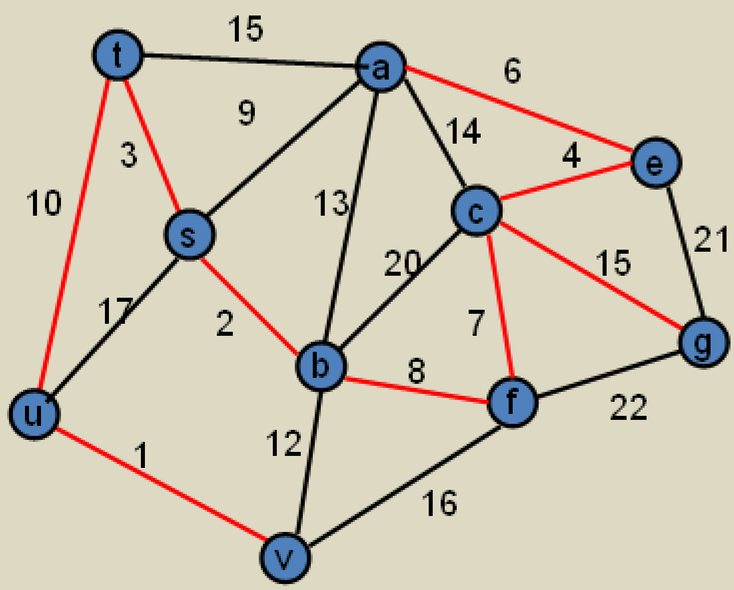
Autumn 2023
Lecture 14
Finishing Minimum Spanning Trees

10/27/2023 CSE 417 1

Announcements

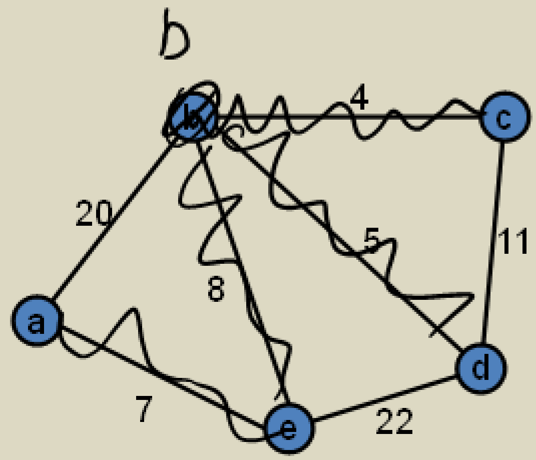
- Midterm, Monday, October 30
  - Closed book, closed notes, no calculators
  - Time limit: 50 minutes
  - Answer the problems on the exam paper.
  - If you need extra space use the overflow page (and the back of a page if necessary)
  - Problems are not of equal difficulty, if you get stuck on a problem, move on.
  - ``Justify your answer'' means give a short and convincing explanation. Depending on the situation, justifications can involve counter examples, or cite results established in the text or in lecture.
- Homework 5, Available, Due November 3
  - Homework 5 is OPTIONAL

Minimum Spanning Tree



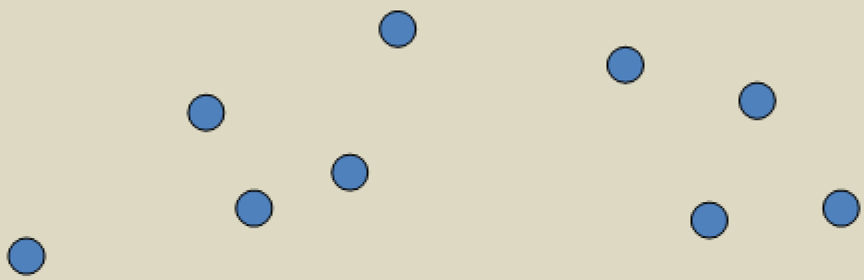
Greedy Algorithms for Minimum Spanning Tree

- Prim's Algorithm: Extend a tree by including the cheapest out going edge
- Kruskal's Algorithm: Add the cheapest edge that joins disjoint components



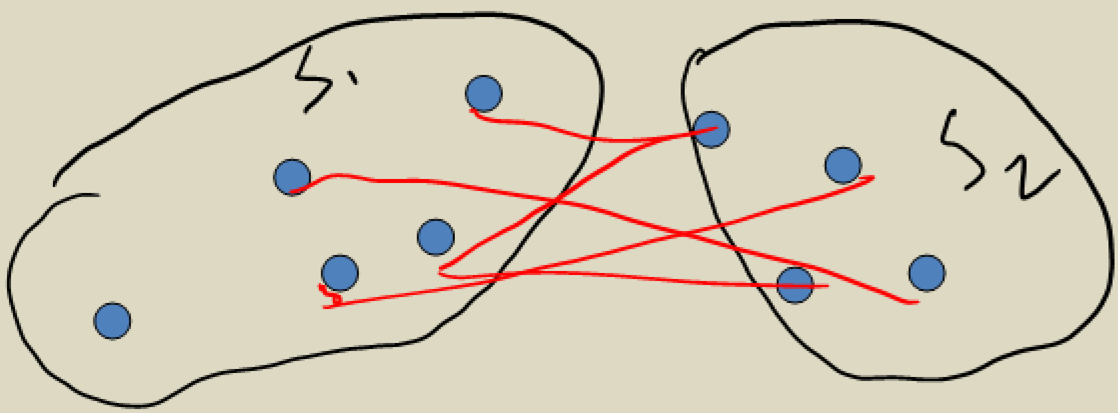
Application: Clustering

- Given a collection of points in an r-dimensional space and an integer K, divide the points into K sets that are closest together

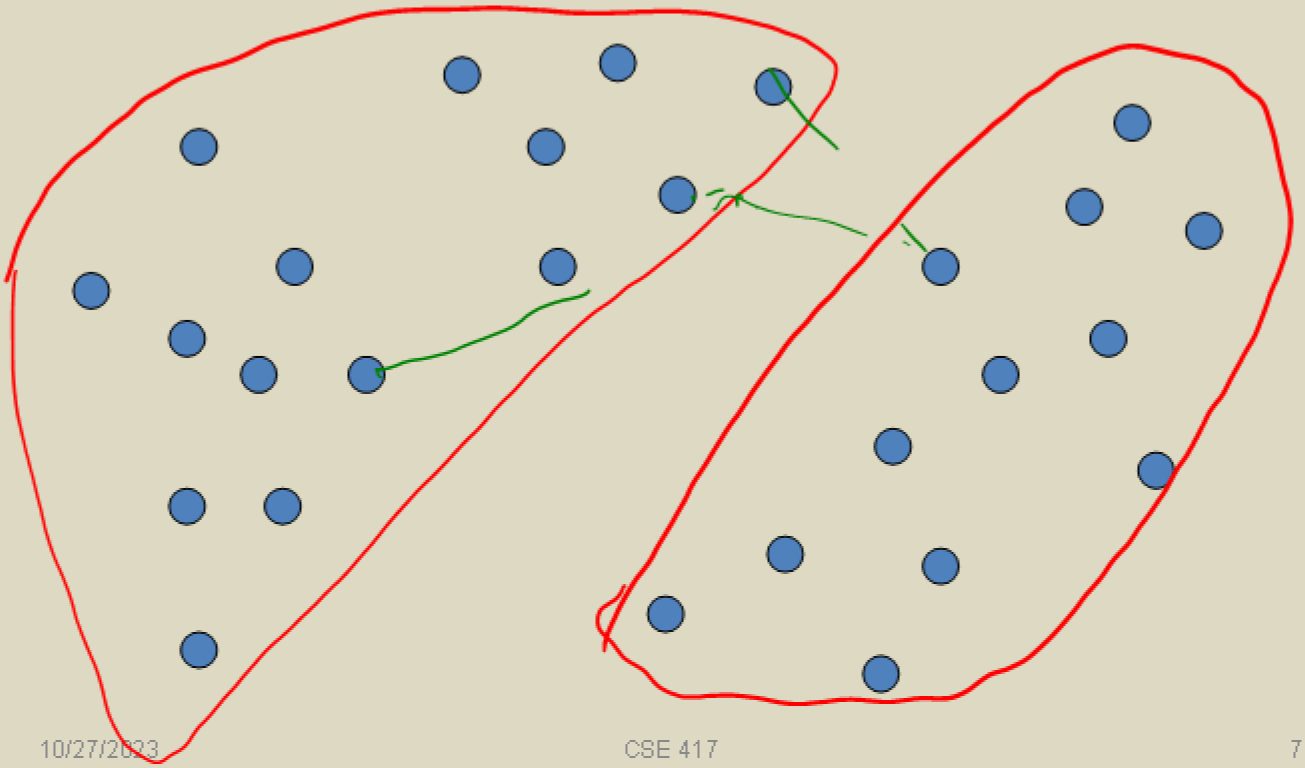


Distance clustering

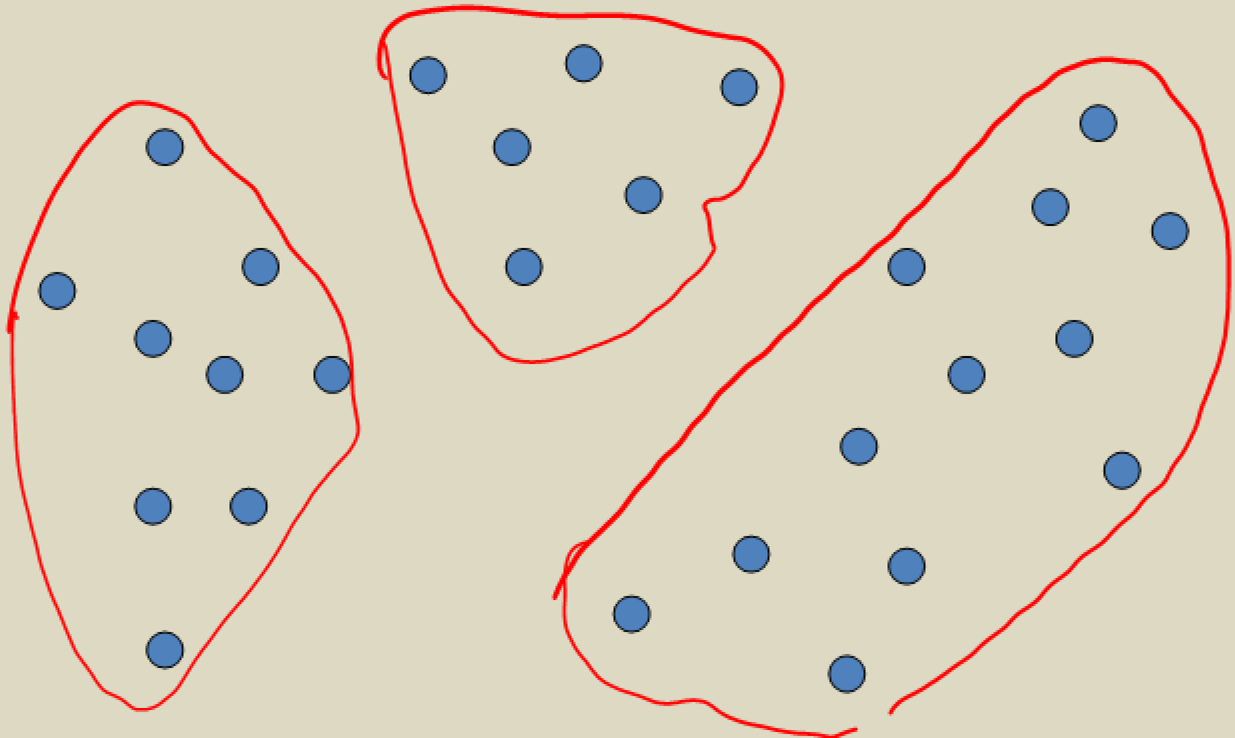
- Divide the data set into K subsets to maximize the distance between any pair of sets
  - dist (S_1, S_2) = min {dist(x, y) | x in S_1, y in S_2}



Divide into 2 clusters

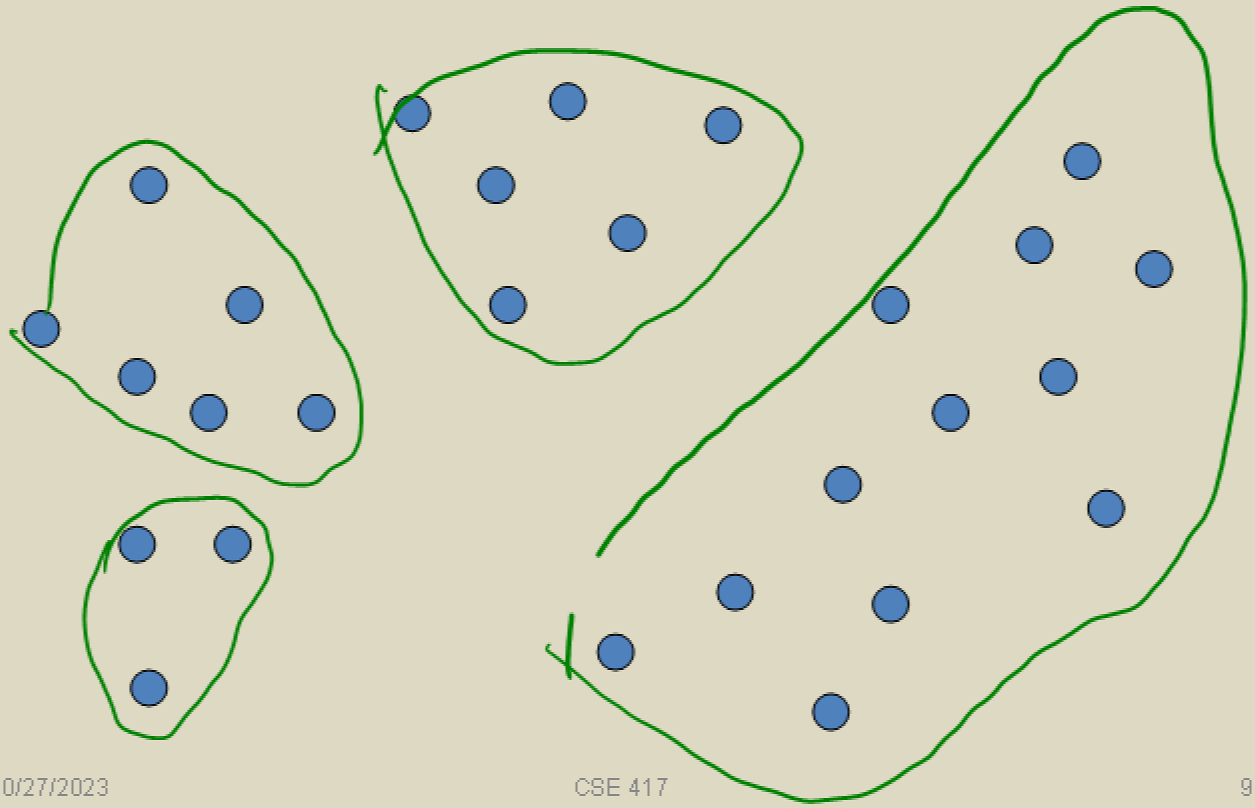


Divide into 3 clusters



10/27/2023 CSE 417 8

Divide into 4 clusters



Hierarchical Clustering

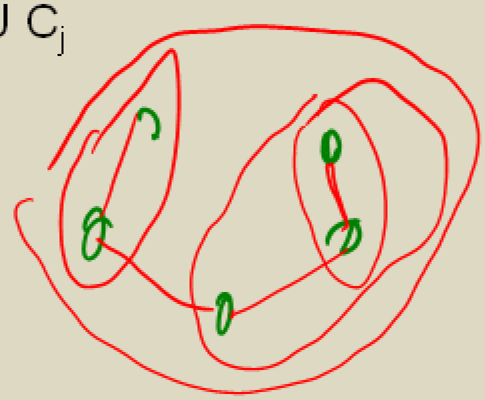
Distance Clustering Algorithm

Let $C = \{\{v_1\}, \{v_2\}, \ldots, \{v_n\}\}$; $T = \{\ \}$

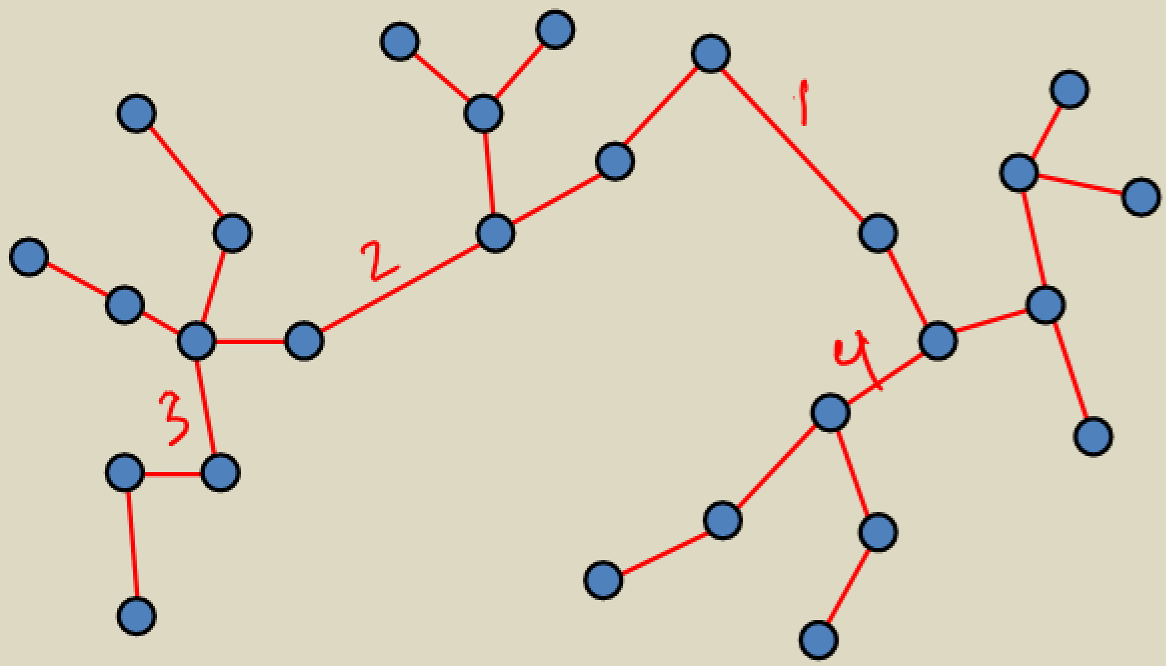
while $|C| > K$

Let $e = (u, v)$ with u in C_i and v in C_j be the minimum cost edge joining distinct sets in C

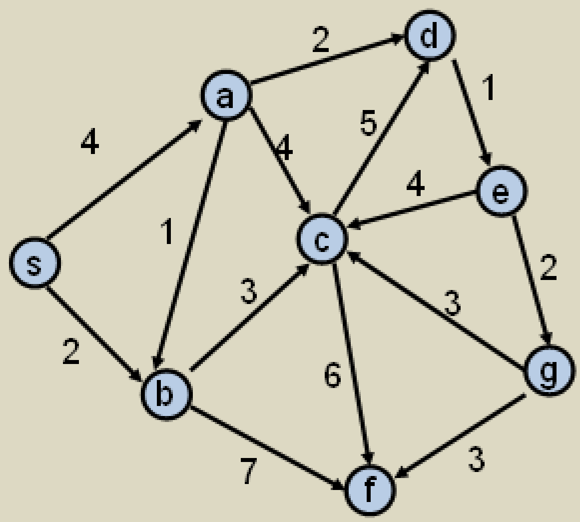
Replace C_i and C_j by $C_i \cup C_j$

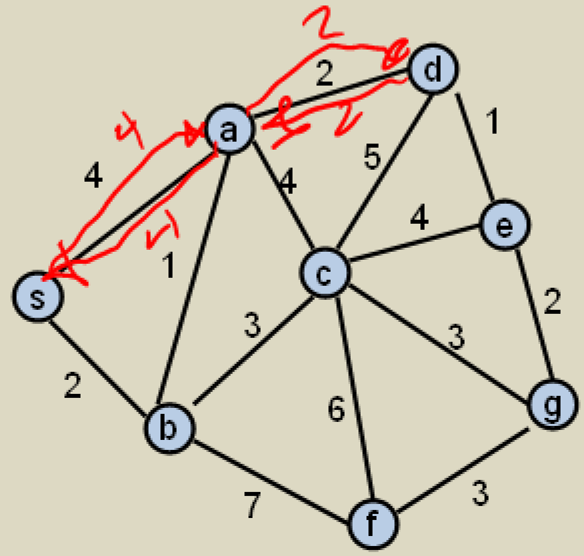


K-clustering



Shortest paths in directed graphs vs undirected graphs

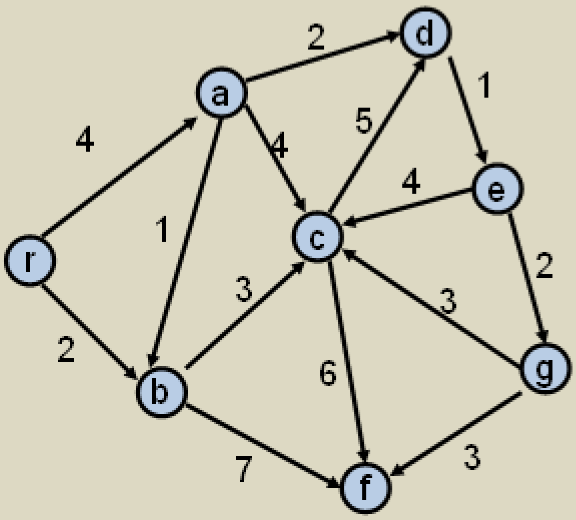




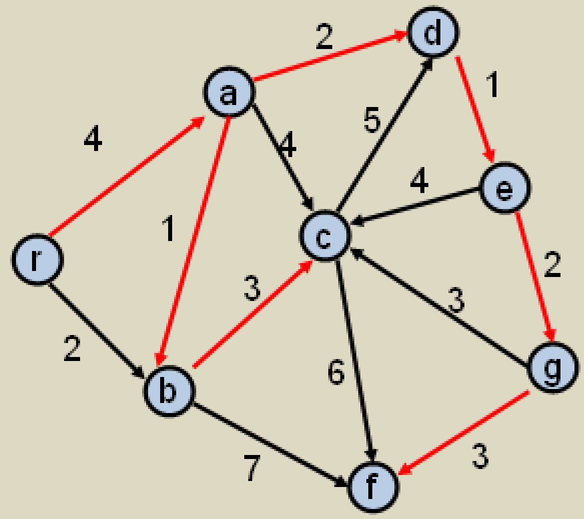
10/27/2023 CSE 417 12

What about the minimum spanning tree of a directed graph?

- Must specify the root r
- Branching: Out tree with root r

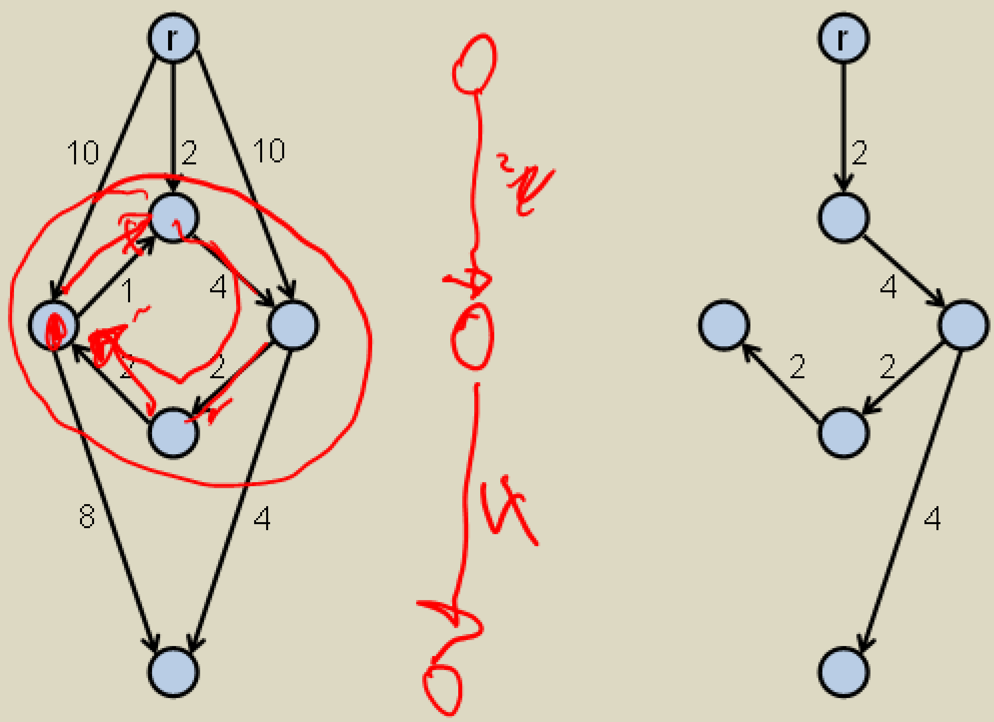


Assume all vertices reachable from r



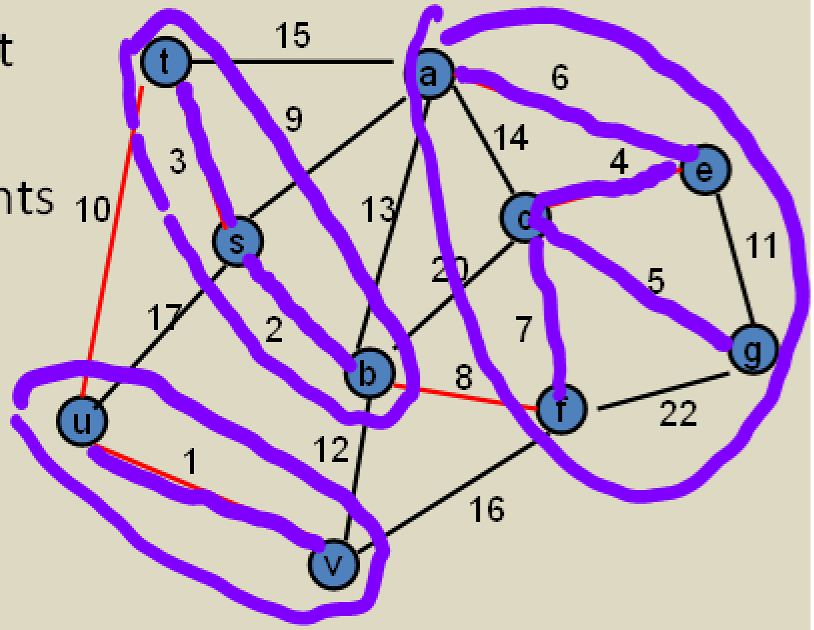
Also called an arborescence

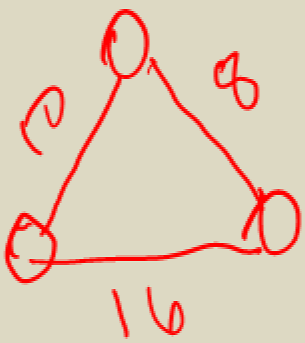
Finding a minimum branching



Another MST Algorithm

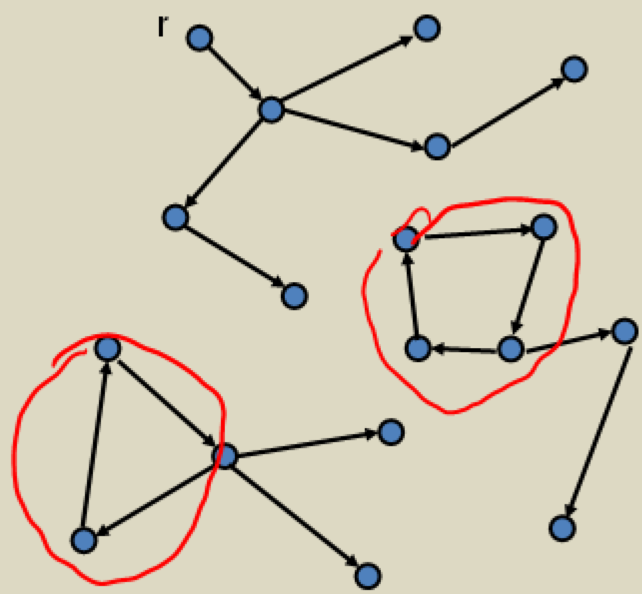
- Choose minimum cost edge into each vertex
- Merge into components
- Repeat until done





Idea for branching algorithm

- Select minimum cost edge going into each vertex
- If graph is a branching then done
- Otherwise collapse cycles and repeat

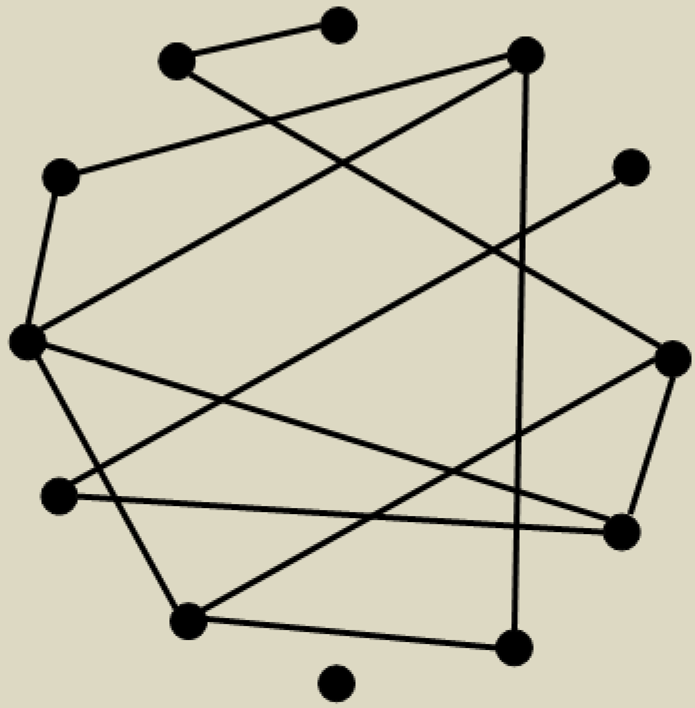


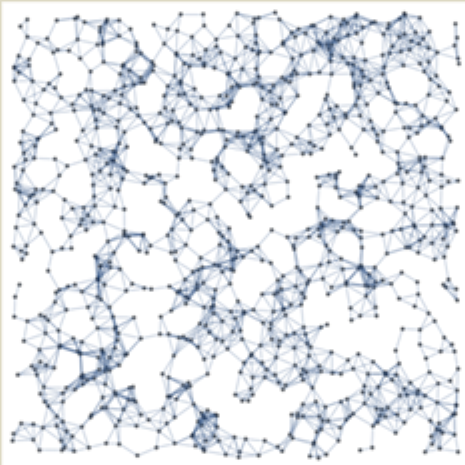
Homework 5: Create a program for coloring random graphs

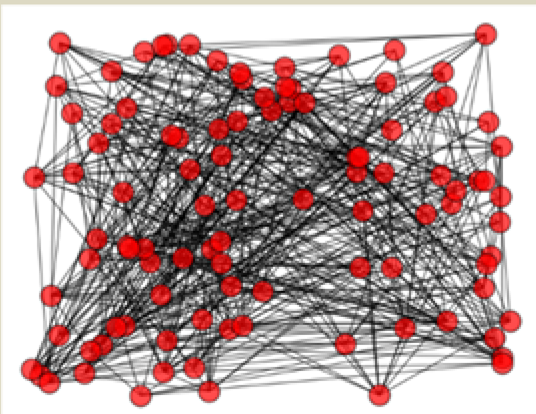
- Optional – this assignment is Just for fun!
- Problem 1: Generate Random Graphs
- Problem 2: Greedy coloring algorithm – first available color
- Problem 3: Low degree first and high degree first heuristics
- Problem 4: Can you find a better coloring algorithm

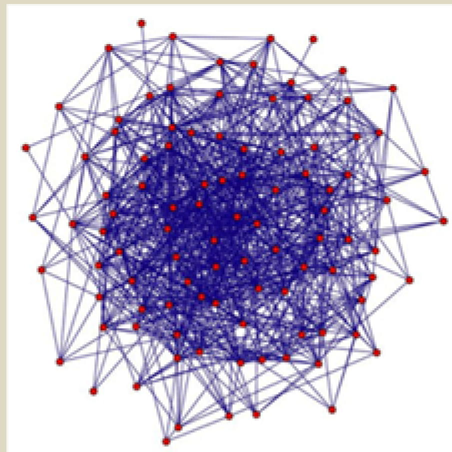
Random Graphs

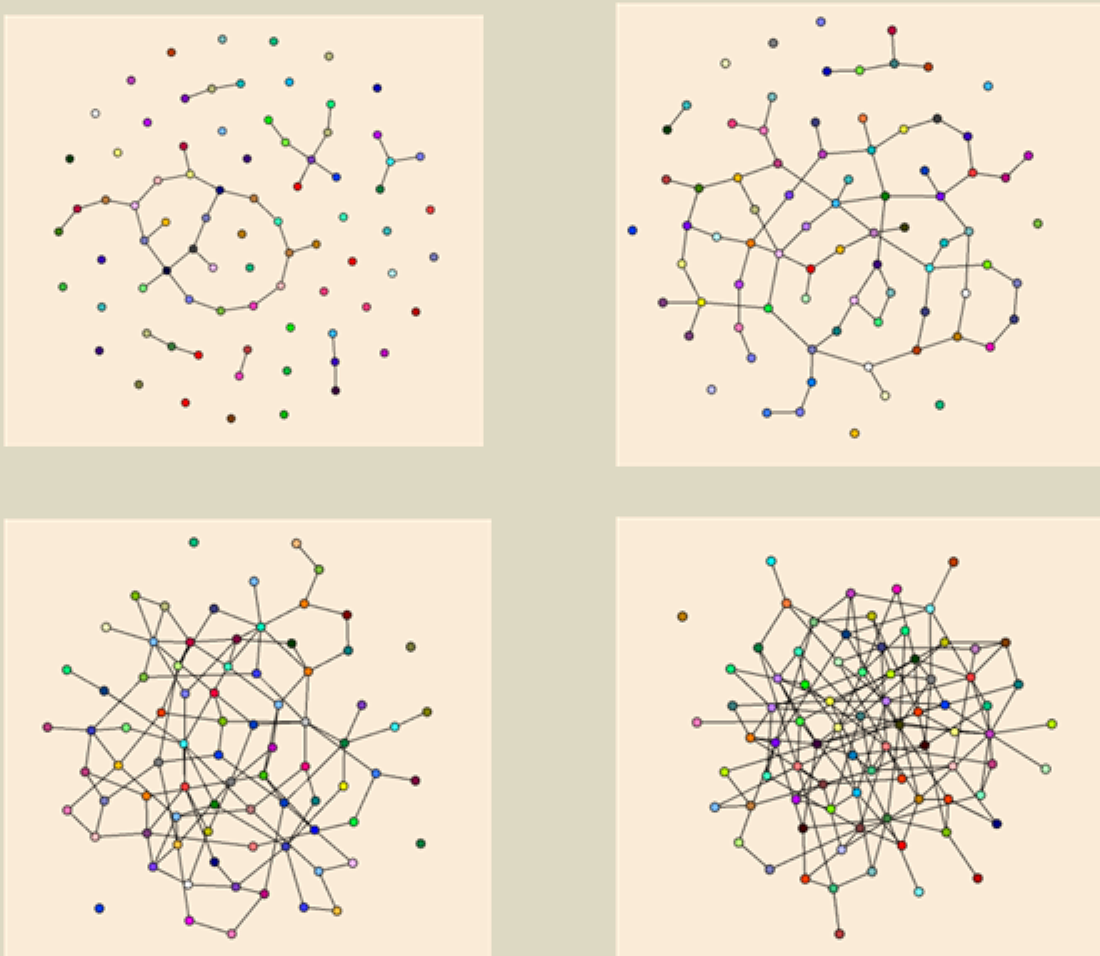
- What is a random graph?
- Choose edges at random
- Interesting model of certain phenomena
- Mathematical study
- Useful inputs for graph algorithms











Model of Random Graphs

- Undirected Graphs
  - Random Graph with n vertices and m edges, G_m
  - Random Graph with n vertices where each edge has probability p, G_p
  - Models are similar when p = 2m / (n * (n – 1))

```
for (int i = 0; i < n - 1; i++)
   for (int j = i + 1; j < n; j++)
       if (random.NextDouble() < p)
           AddEdge(i, j);
```

Coloring Random Graphs

- Chromatic index of a graph G, $\chi(G)$ – minimum number of colors needed to color G
- Mathematical question, given a graph g chosen at random from $G_p(n)$, what is the expected value of $\chi(g)$
- Graph coloring is NP complete – suggesting that it may be hard to determine $\chi(g)$
- There is a fairly large gap between the heuristic results and the theoretical value
- There are a number of ties with theoretical physics