For Friday

- Chapter 9, section 6
- Program 3 due

Research Paper

- Any questions?

Program 3

- Any questions?

All Shortest Paths

- In sparse matrix, just run Dykstra's algorithm for each vertex
- In dense graphs, use a matrix and use an algorithm similar to Warshall's algorithm
- Computes all shortest paths in $O(V^3)$ time
- To determine actual path, need an additional matrix.

Floyd's Algorithm

```
for (y = 0; y < V; y++)
    for (x = 0; x < V; x++)
        if (a[x][y])
            for (j = 0; j < V; j++)
                if (a[y][j] > 0)
                    if (!a[x][j] ||
                        (a[x][y] + a[y][j] < a[x][j]))
                        a[x][j] = a[x][y] + a[y][j];
```

Minimum Spanning Tree

- Useful to solve problems like: what's the cheapest way to connect a set of points to each other
- Can be solved using a greedy algorithm
- So we can use best-first search (search using a priority queue)
- Applies to undirected graphs

Minimum Spanning Trees

- Prim's algorithm

Select a node to put in the spanning tree
while some nodes are not in the tree
    select the shortest edge connecting a node
        in the tree to a node NOT in the tree
    add that edge and node to the tree
end while

Minimum Spanning Trees

- Kruskal's Algorithm

while not all nodes are in a single tree
    select the shortest edge with at least one
        end node not currently in a tree or
        connecting to trees to each other
    add that edge and its end nodes to the tree
end while

Graph Searching

- Depth-first
- Breadth-first
- Best-first

Applications of Searching

- Connectivity in an undirected graph