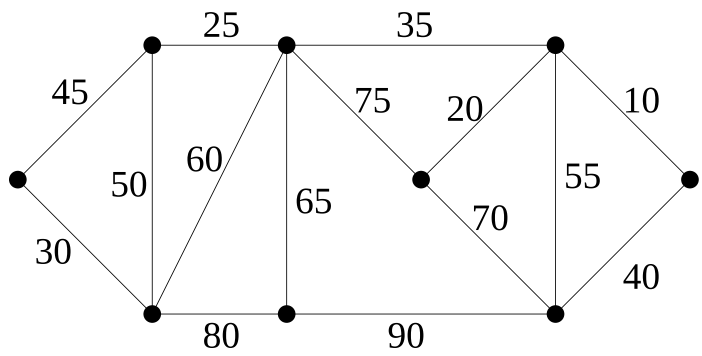
1. Consider the following undirected graph.



   (a) Assume that you use Kruskal's algorithm to find the Minimum Spanning Tree (MST). Show the order that the edges are included in the MST. Just list the edges in order by their weights.

   (b) Assume that you use Prim's algorithm to find the Minimum Spanning Tree, using the leftmost vertex as the source. Show the order that the edges are included in the MST. Just list the edges in order by their weights.

   (c) Assume that you use Dijkstra's algorithm to find the shortest paths from the leftmost vertex. Show the order that the edges are included in the shortest paths tree. Just list the edges in order by their weights.

2. Assume we modify Dijkstra's algorithm so that rather than finding the shortest paths, we try to find the longest, *acyclic* paths in a weighted, directed graph. We initialize the distances in d to $-\infty$, change the $<$ to a $>$ (in the relaxation statement), and extract the maximum value from outside with respect to d. The new algorithm has the same (worst case) time complexity.

```
for each vertex v ∈ V[G] do
    d[v] ← −∞; π[v] ← NIL
end for
outside ← V[G]

d[1] ← 0
while outside ≠ φ do
    u ← Extract_Max(outside,d)
    for each v adjacent to u do
        if v ∈ outside and d[u] + W[u,v] > d[v] then
            d[v] ← d[u] + W[u,v]
            π[v] ← u
        end if
    end for
end while
```

   Will this find the longest, acyclic paths from the source? Justify.

3. Consider the following three arrays of data:

```
MatrixValues:   1 2  1 1 3 1  2 1 3 1  1 3 1  1 3 1 2 1  1 2 4  3 1 4

RowStart:  1 3 7 11 14 19 22 25

ColumnIndices:  2 3  1 3 4 5  1 2 5 6  1 2 5  2 3 4 6 7  3 5 7  4 5 6
```

This is an example of a way to store an $n \times n$ matrix A:

- `MatrixValues` is a list of the nonzero entries in the matrix A. The nonzero values in row 1, are followed by the nonzero values in row 2, are followed by the nonzero values in row 3, etc.
- `RowStart` are pointers to where the values in each row of A start. The first value of RowStart is always 1; the next value is 1 plus the number of nonzero values in the first row; the next value is 1 plus the number of nonzero values in the first two rows; etc. The final value in RowStart is 1 plus the total number of nonzero values.
- `ColumnIndices` represents the column indices of the nonzero entries of the matrix A. The columns of the nonzero values in row 1, are followed by the columns of the nonzero values in row 2, are followed by the columns of the nonzero values in row 3, etc.

(a) Given the data above, write down the matrix A in standard format.

(b) This matrix represents a directed, weighted graph. Nonzero values represent an edge and gives its weight. Draw this graph.

(c) For a graph with n vertices and m edges what is the size of the input (in Θ notation)?

(d) Why might one want to store a matrix in this format?

(e) Give pseudo code to convert a matrix in standard format into this format.

(f) Give pseudo code to convert a matrix in this format into standard format.

4. **Challenge Problem. Will not be graded.** Assume that we have four cities on the four corners of a unit square. Using the Euclidean distance, the Minimum Spanning Tree (MST) has length 3. Assume that you are allowed to add cities, which must be included in the MST, then you can actually create a graph with smaller MST. For example, if you put a new city in the center of the square, the MST will have length only $2\sqrt{2} \approx 2.8$. Where should cities be added to minimize the length of a MST (that includes all four original cities), what is the MST, and what is its length?