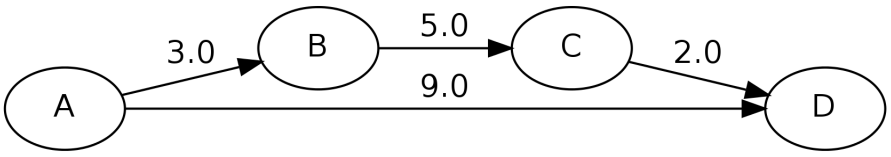
Graph Algorithms

Not all of the questions below will be covered in precept. Solutions will be posted after precept.

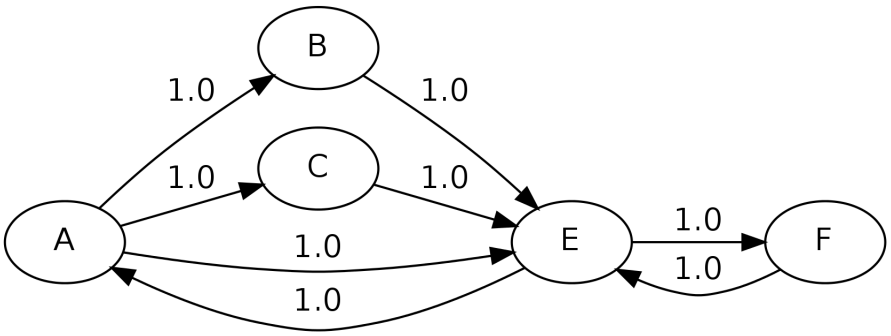
1. For the graph below, if we use the eager version of Prim's algorithm (with code modified to handle directed graphs) starting from vertex A, what two vertices are on the PQ immediately after all of A's outgoing edges are processed? What are their priorities?



2. For the graph above, what two vertices are on the PQ after the starting vertex is relaxed in Dijkstra's algorithm? What are their priorities?

3. As succintly as possible, state the main difference between Dijkstra's algorithm and the eager version of Prim's algorithm? Give your result in terms of how the algorithm makes decisions, not in terms of the problem it's intended to solve. Hint: What will be the priority of vertex C when it's put on the PQ?

4. From class, we know that every vertex is relaxed exactly once. In the graph below, if we run Dijkstra's algorithm, why is it true that vertex A is only put into the priority queue exactly once? In other words, when E is relaxed, why doesn't A go back into the priority queue?



5. Given a graph where all edge weights are exactly 1. Consider the following proposition about Dijkstra's algorithm. All vertices on the PQ have either priority k or k+1. Is this proposition true?

6. In a graph where all edges have equal weight, does Dijkstra's algorithm differ from BFS in any meaningful way? Why or why not?

7. If we wanted to solve the shortest paths problem for an *undirected* graph, how would we do this if all edge weights were positive? Why are things more complicated if any edge weights is negative?

8. True or false: If we run Dijkstra's algorithm from an arbitrary vertex on a weighted graph, we get a minimum spanning tree. If true, why? If not, give a counter-example containing no more than 3 vertices.

9. In 8puzzle, we used a graph search algorithm called A*. In this algorithm, we were finding a shortest path from x to y.

   (a) Is the 8puzzle graph weighted or unweighted?

   (b) Is the entire 8puzzle graph constructed at the time the program starts?

   (c) Is the A* algorithm equivalent to DFS, BFS, Dijkstra's, or none of these?

10. What are our three shortest paths algorithms for weighted graphs, and when should you use each?

Seam Carving

The image below is 5 columns by 6 rows with artificially small colors. Each color is listed as (R, G, B).

```
(3,  3,  0)     (1,  0,  3)     (0,  1,  0)     (2,  1,  3)
(3,  2,  3)     (1,  0,  1)     (2,  0,  2)     (3,  3,  3)
(3,  1,  3)     (1,  2,  0)     (2,  2,  1)     (1,  1,  3)
(0,  2,  0)     (3,  0,  3)     (3,  0,  2)     (2,  0,  1)
(2,  2,  3)     (1,  2,  0)     (3,  2,  2)     (3,  3,  2)
```

1. The array below has most of the energy values calculated with the colors given above (which have artificially low values), fill in the blanks below by calculating the energy for (0, 0) and (2, 1). Keep in mind we number our points by x, y coordinates!

```
     30   28    7   19
18   19   14   17    6
 7   23         7   23
14    7   20   26    7
```

2. Find the vertical seam using your intuition and circle it.

3. Use the cumulative energy technique to calculate the horizontal seam and circle it. For your convenience, a copy of the energy values is shown below.

```
     30   28    7   19
18   19   14   17    6
 7   23         7   23
14    7   20   26    7
```