CS 483
Homework 2
due Wednesday, June 17

1. The following is a nonsense recursive algorithm:

```
skrunk(array a, int high)
{
   if (high > 0)
   {
      val = a[high/2]
      for (i = 0; i < a.length; i++)
         if (a[i] == val)
            print("whoopie!")
         else
            print("oops!")
      skrunk(a, high/2)
   }
}
```

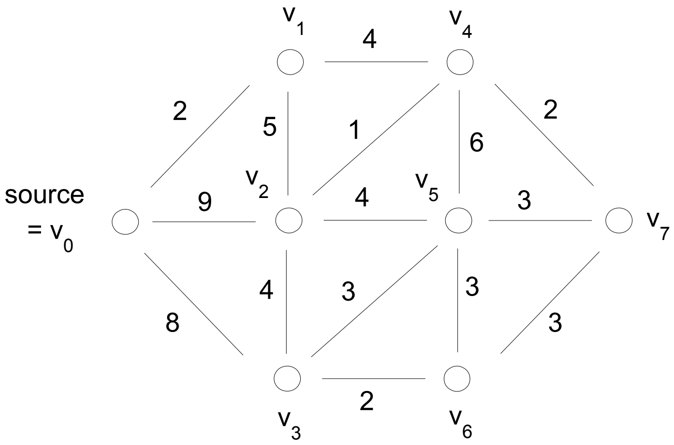
The initial call is to skrunk(a, a.length - 1)

Choose a representative operation and state your choice. Compute c(n) (the number of times the operation is performed for an array a with n = a.length) and give its big theta class. (You do not have to give any proof.)

2. Skip problem #2.

3. Dijkstra's algorithm for shortest distances from vertex v_0 is being performed on the weighted graph at the right. At the present moment the set S of completed vertices is S = {v_0, v_1} and the distances computed are:

v_0.d = 0, v_1.d = 2, v_2.d = 7, v_3.d = 8, v_4.d = 6, v_5.d = ∞, v_6.d = ∞, v_7.d = ∞.



Perform the next step (i.e. the next pass through the main loop) in Dijkstra's algorithm. Show what the set S is after this step and show what distances v.d have been changed and what their new values are.