

```

(a):
len(a)
n <= 1:
return a

= 1 * n // 3
= 2 * n // 3

= mergesort3
le = mergesort3
= mergesort3

```

```

de3: T(m) <= 1
n merge3(left,

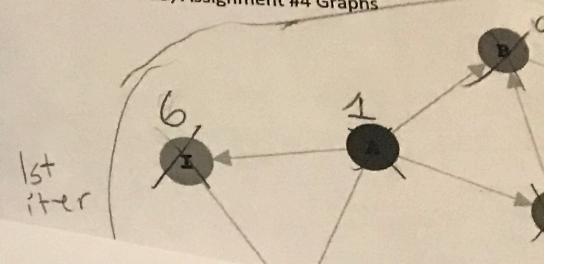
```

er theorem to ge  
what case does

asymp  
Case 1

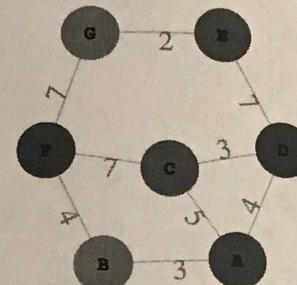
continues on th  
recursion tre

Names: Cattlin Walker



### Assignment #5, Dijkstra's Shortest Path Algorithm

- Compute the shortest path from D to all other vertices.



A	B	C	D	E	F	G
4	7	3	0	7	10	9

You must show your work below to receive full credit. Specifically, show your candidate edges (alphabetically) for each iteration of Dijkstra's Shortest Path Algorithm.

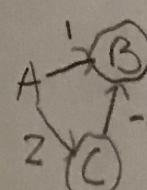
Edges	Path Lengths
Iteration 1	
D → A	4
D → C	3 *
D → E	7
Iteration 2	
C → A	8
C → F	10
D → A	4 *
D → E	7
Iteration 3	
A → B	7 *
C → F	10
D → E	7
Iteration 4	

B → F	11
C → F	10
D → E	7 *
Iteration 5	
B → F	11
C → F	10
E → G	4 *
Iteration 6	
B → F	11
C → F	10 *
E → F	10

- Why doesn't Dijkstra's work on graphs with negative edges?

If there were negative edges then there could possibly be a path that we haven't considered that is shorter. With Dijkstra's we pick the shortest path every time and build upon that so by always taking the shortest path we are assuming that all of the other possible paths are longer because a path that already isn't the min. plus a pos number is already longer. But if there is a negative edge then that could potentially be shorter.

Score from AutoGrader: 100%

ex.   
Dijkstra's would assume A → B is shorter but actually A → C → B is shorter.