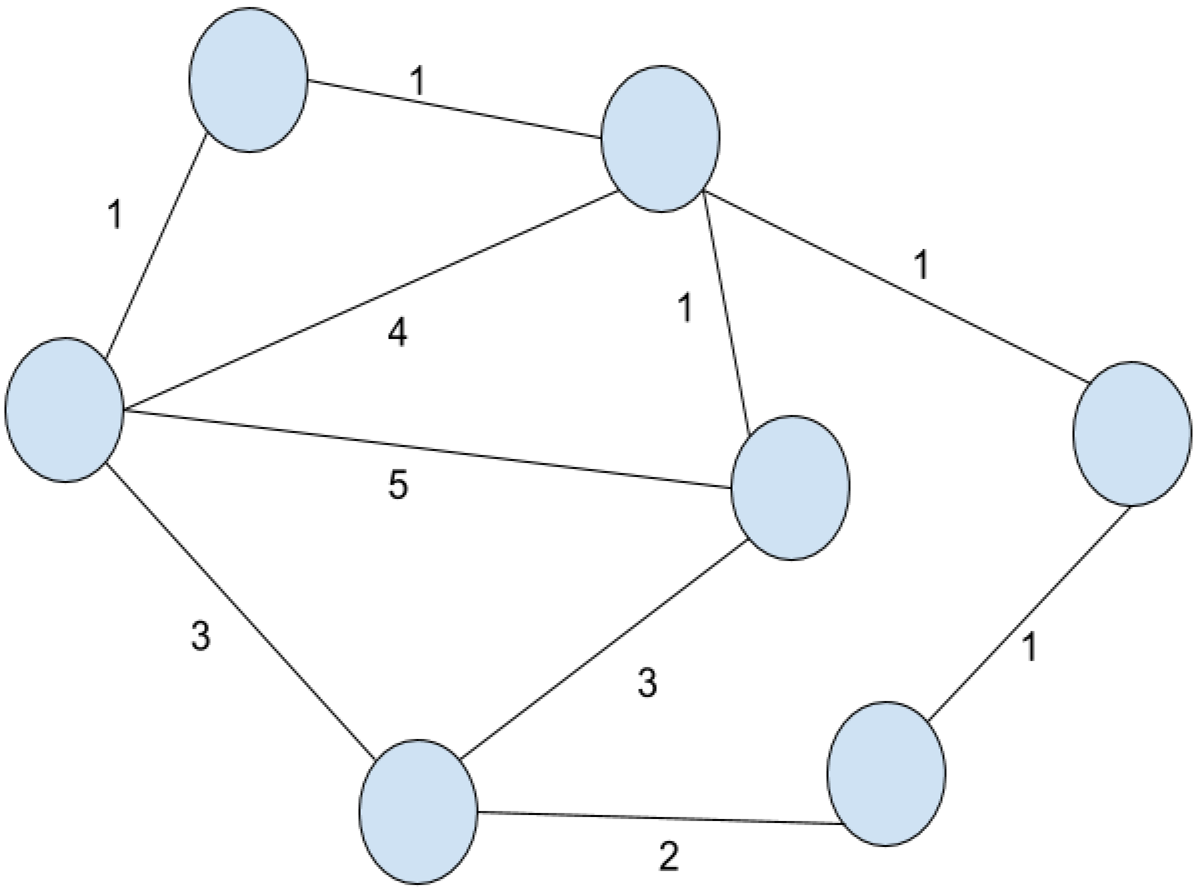
Recitation 9 (10/31- 11/4)

Dijkstra's Algo

Finds the shortest path to all nodes from a starting node, where the edges are weighted.

Walk through examples on the graph below.



Returned: List of nodes with their distances

Talk about storing previous values if we only cared about the distance from a start node to the end node.
Can also stop finding distances once you have reached the end_node

NOTE - no greedy algorithm can solve the problem if some edges had negative weights

Exchange Argument for Minimizing Max Lateness

Idea: If we have a schedule with no idle time and with an inversion, getting rid of the inversion will not increase the max lateness of the schedule.
Schedules with no inversions and no idle time have the same max lateness.

Given the schedule below where the deadline of j comes before the deadline of i:
The top schedule has an inversion.

Obviously j will finish earlier so it will not increase the max lateness, but what about i? Now i is super late.

i's lateness is f(j) - d(i) where f(j) is the finish time of j in the orig schedule.
$f(j) - d(i) < f(j) - d(j)$ so the swap does not increase maximum lateness.

Explicit example showing this:
d(a) = 2 t(a) = 1
d(b) = 5 t(b) = 6
d(c) = 8 t(c) = 3

Orig schedule with inversions:

__a__|____c____|______b_______

Schedule without inversions:

__a__|______b_______|____c____

Schedule 1 max lateness: 5
Schedule 2 max lateness: 2
NOTE: Examples do not suffice as proofs. But they are good for understanding purposes.