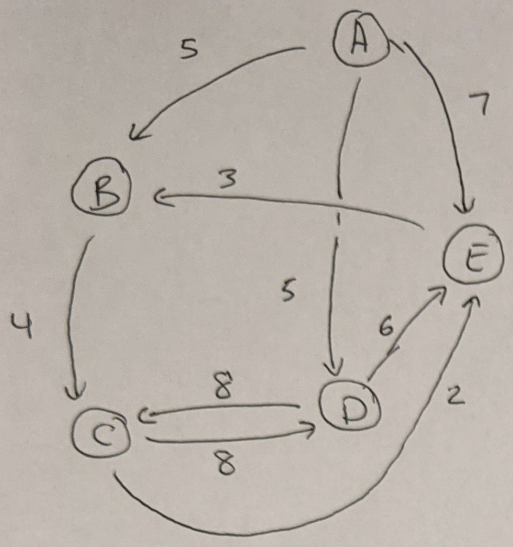
Haitai Ng.



(A) Distance of route ABC
5 + 4 = 9

(B) The number of trips starting at C and ending at C with maximum 3 stops
- C DC (2 stops)
- CEBC (3 stops)
- ~~CDEBC~~ (4 stops)

(C) The number of ~~different~~ routes from C to C with a distance of less than 30
- CDC = 16
- CEBC = 9
- CEBCDC = 16+9 = 25
- CDCCEBC = 25
- CDEBC = 21
- CEBCEBC = 18
- CEBCEBCEBC = 27

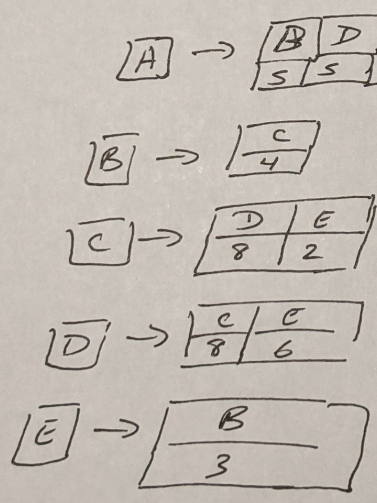
(D) shortest route (in terms of travel) from A to C
→ use Dijkstras Algorithm
→ 9.

- graphs can be represented using either Adjacency matrix of Adjacency List.

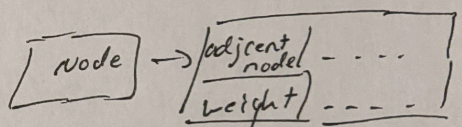
Adjacency Matrix

	A	B	C	D	E
A	0	5	0	5	7
B	0	0	4	0	0
C	0	0	0	8	2
D	0	0	8	0	6
E	0	3	0	0	0

Adjacency List



* For this problem I will be using an adjacency List data structure to represent the graph.



- Adjacency Lists (directed graph with weight)
- Depth first search to find all paths from source to destinations (modified depth first search)
- Recursive / NP to find cyclic routes
  - modified depth first search can find paths.
  - how to recursively find all permutations where C = # of stops & C = max Allocated distance
- Dijkstra's Algorithm for shortest path between two nodes.