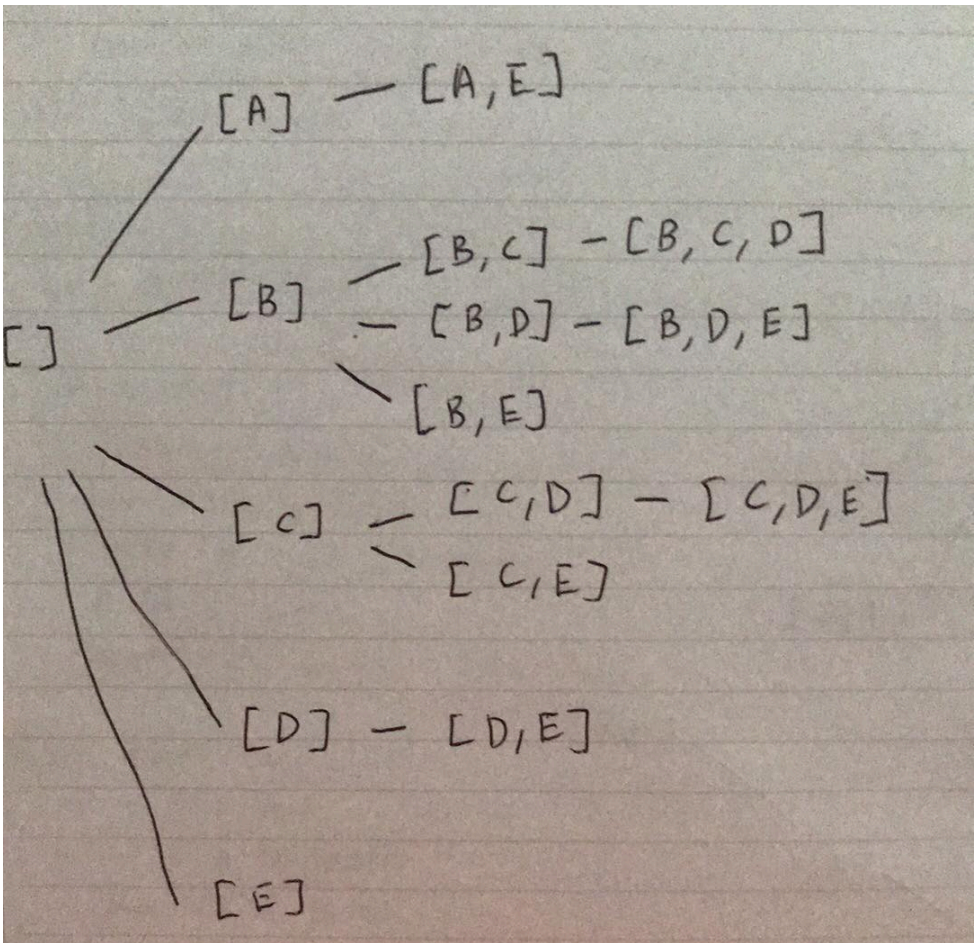
Danny Tan

AI Fall 2018

HW # 2

Problem 1:

1a.



1b.
[], [A], [A,E], [B], [B,C], [B,C,D]

1c.
[], [A], [B], [C], [D], [E], [A,E], [B,C], [B,D], [B,E], [C,D], [C,E], [D,E], [B,C,D]

2.

- Is the state space a tree?

    - Yes the state space is a tree
- Give an upper bound on the depth of the state space.
    - N+1 in the worst case if it has all N objects
- What is the branching factor?
    - The number of nodes whose weight is less than the max weight minus the current node's weight and comes after alphabetically
- Is the depth of the shallowest goal known in advance?
    - No because the depth is not known before until you parse through all of the cases. The depth changes according to the maximum weight and minimum value.

3a.

The start state should start with a node for every vertices. For each node, its children should be the adjacent list of vertices as its parent or ancestor. The answer should be one of the leaf node that has a depth of K or 4 in this case.

b.

The depth of the state should be K which is 4. Therefore, DFS should be used since the goal state is at the leaf. On other hand, BFS will have to a lot of cases of the before it gets to the leaf node since it is moving through the tree horizontally.

c.

i. The depth of the state space should be Min(Q,K).

ii. The branching factor is Q.

iii. Min(Q^2, QK)