SEARCH TREE METHODS

CS340

Search Tree Methods

- How to solve hard combinatorial problems?
- Exhaustive search (enumerating all candidate solutions and identifying the one with a desired property) takes too long.
- The set of all possible choices is called the “state space”.

Searching the state space

- Widely used in Artificial Intelligence
- Essentially a DFS
- Don’t get your hopes up
  - The state space probably needs to be narrowed considerably

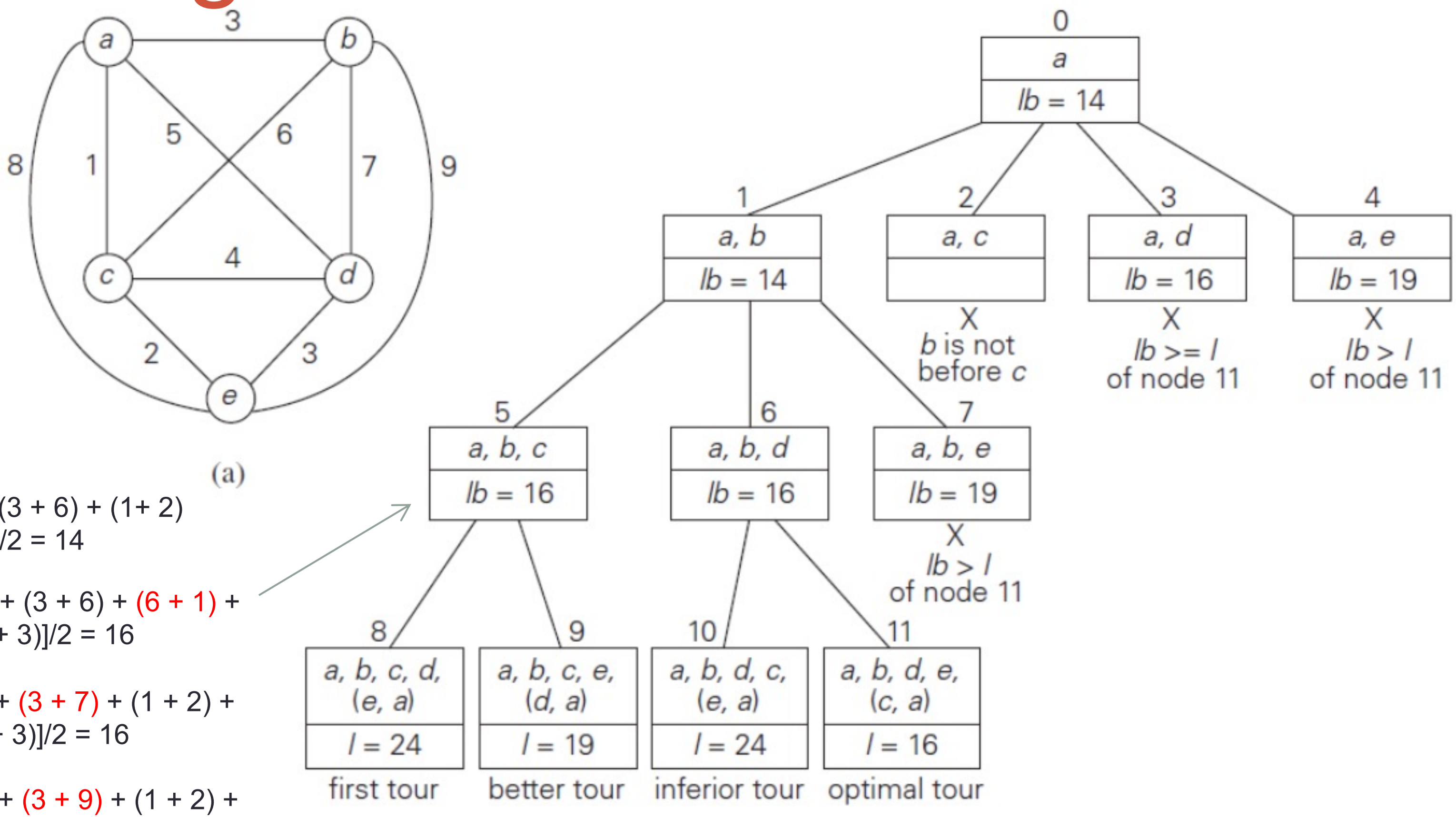
- Eg. Clique
  - The state space is all subsets of size k.
  - Search the state space to see if it contains a clique.
  - Is it possible to increase speed by thinking about the problem?

Narrowing the state space

- For finding the max clique in a graph:
    - It is easy to find an upper bound on the size of a clique.
    - Vertices of low degree cannot be part of a large clique.
    - Removing low degree vertices might break up the graph.
    - Start by finding a small clique of size k’ (which is much easier than finding a large clique), and stop considering vertices with degree lower than k’.
    - Keep track of the largest clique found, and stop considering cases where addition of all possible remaining vertices will not make a larger clique.
- The case where we stop considering is called “pruning”

Narrow the State Space for Traveling Salesman?

Traveling Salesman



(a)

Init LB = [(1+ 3) + (3 + 6) + (1+ 2) + (3 + 4) + (2 + 3)]/2 = 14

ABC LB = [(1+ 3) + (3 + 6) + (6 + 1) + (3 + 4) + (2 + 3)]/2 = 16

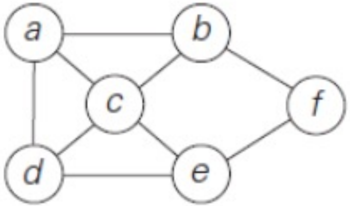
ABD LB = [(1+ 3) + (3 + 7) + (1 + 2) + (3 + 7) + (2 + 3)]/2 = 16

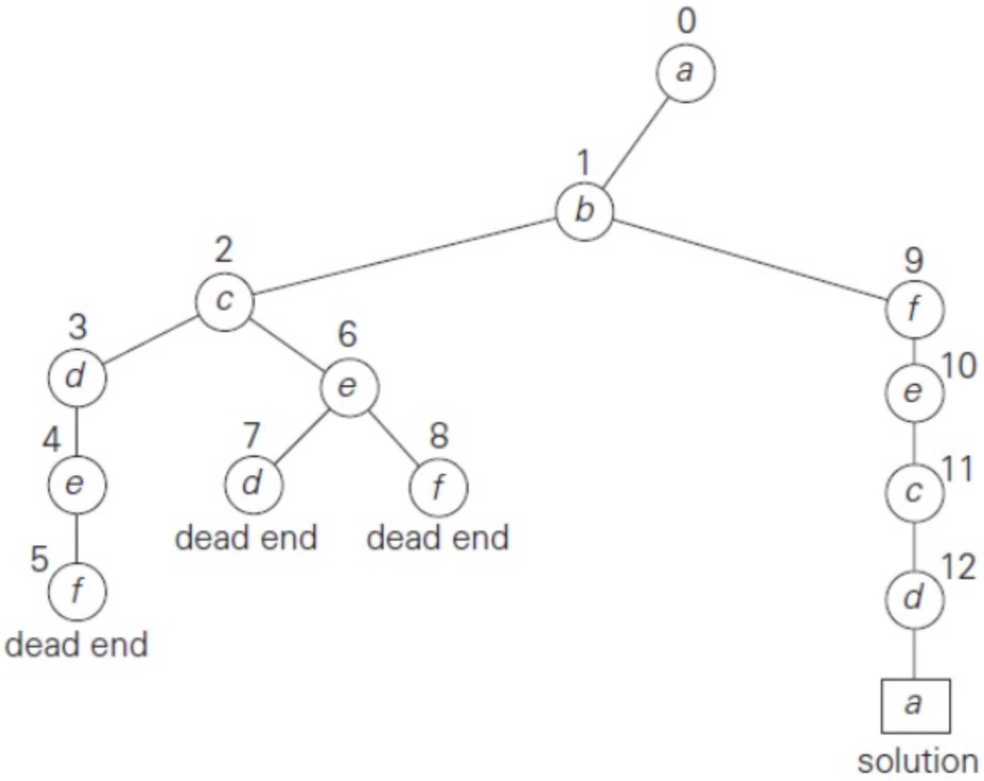
ABE LB = [(1+ 3) + (3 + 9) + (1 + 2) + (3 + 4) + (2 + 9)]/2 = 19

Backtracking

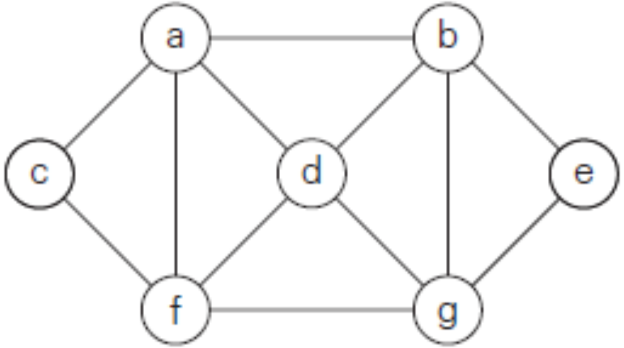
- When we have no way of ranking paths down the tree
    - One partial solution not known to be better than another partial solution

Hamiltonian Circuit

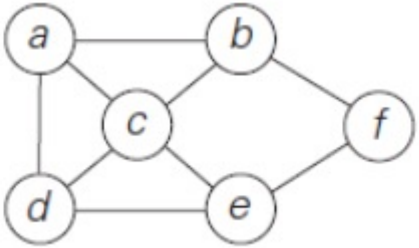




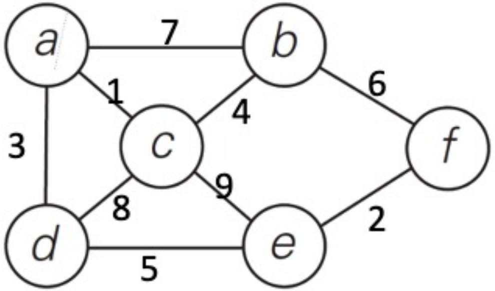
Hamiltonian Circuit



BFS/DFS



Prim/Dijkstra



Subset Sum

3 items, weights 5,1,3. Total weight for basket: 5

item 3 (w=3)	0					
item 2 (w=1)	0					
item 1 (w=5)	0					
	0	0	0	0	0	0
weight	0	1	2	3	4	5