NAME: YASIN AYDIN

CS 404 – Artificial Intelligence
Spring 2019

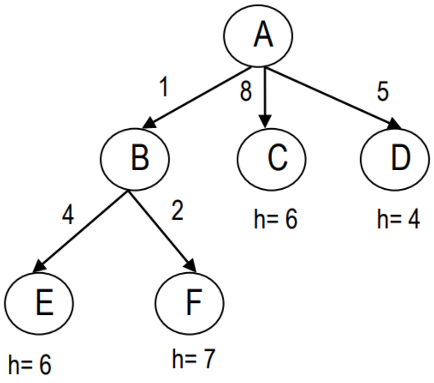
HW3–Local Search – Adversarial Search

75pts

1) **5pts –** Give the name of the algorithm that results when you do a local beam search with k = 1.

Answer: **Hill-climbing algorithm**

2) 30pts - Consider the following **partial** search tree (we are in the middle of the search), where each edge is labeled with the cost of the corresponding operator and the leaves (fringe nodes) are labeled with the value of a heuristic function, h, estimating the remaining cost to the goal. Which node will be expanded next by each of the following search methods? Give a very small explanation or show your work.

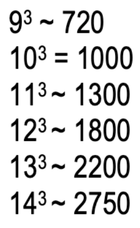


1. Uniform-Cost Search: The next node that will be expanded is 'F' since the f(n) = "cost from start to n" is the lowest among others (F). (f(D)=5, f(C)=8, f(E)=5, f(F)=3)
2. Greedy Best-First Search: The next node that will be expanded is 'D' since the next smallest 'h' in the fringe is 'D' with h=4.
3. A* Search: The next node that will be expanded is 'D' since the f(n) = g(n) + h(n) is the lowest among others (9). (f(D)=9, f(C)=14, f(E)=11, f(F)=10)

NAME: YASIN AYDIN

3) **10pts** A heuristic results in exploring N=180 nodes and finds the solution at depth d=2. What is its effective branching factor? Give an **approximate** answer, but you must show your work.

Hint:



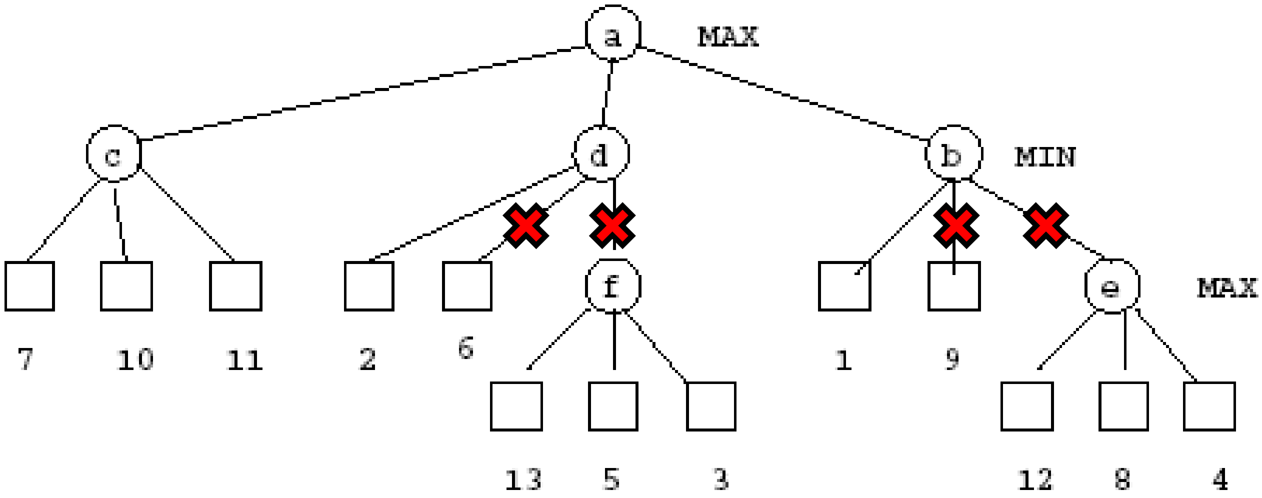
$9^3 \sim 720$
$10^3 = 1000$
$11^3 \sim 1300$
$12^3 \sim 1800$
$13^3 \sim 2200$
$14^3 \sim 2750$

In order to check 180 nodes within depth 2, we need a branching factor that satisfies 1 + b + b^2 >= 180 and the smallest number that satisfies this is 13 with 13x13 + 13 + 1 = 183. **So, the effective branching factor 13.**
We solve this equality with the given formula too (b^(3) − 1)/(b − 1) >= 180 which gives us same result branching factor 13.

NAME: YASIN AYDIN

4) 30pts - Game Playing

Using the following Minimax tree, answer the following questions:



a) 5pt - What score is guaranteed for MAX?

Answer: 7

b) 15pt - Indicate **all the nodes** that are pruned using alpha-beta pruning? You can use the node name or values to indicate.

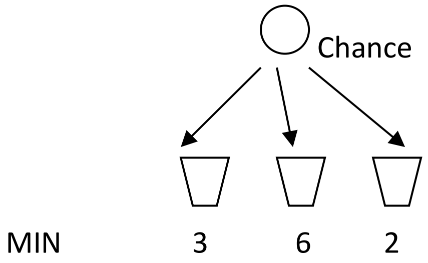
Answer: The pruned branched are shown on the tree and the nodes **are 6,f,9,e.**

c) 5 - True or False: If Max uses alpha-beta pruning in Minimax, can s/he miss the chance of a better play (if s/he did't prune)? Assume a perfect opponent.

Answer: **No, s/he doesn't miss anything because pruning eliminates only the unnecessary nodes which will not be used.**

NAME: YASIN AYDIN

d) 5pt - What is the expectimax value for the following chance node (circle)? Assume equal probability for each of the chance outcome and the given expectimax values for the MIN node.



Expectimax = (3 * 1/3) + (6 * 1/3) + (2 * 1/3) = 3.67

*) For those who have requested extra study questions, other good questions to work on (from the topics we covered) are: AIMA 3rd ed: 4.9 (topic not covered, but in the slides) 5.12, 5.15, 5.18, 5.19, 5.21,