CS 404 – Artificial Intelligence
HW 2 – Blind Search – AIMA– Chp. 3

75pt

Late homeworks accepted for 2 days (no penalty in the first late day; -10pts off when late for 2 days)

Please type your answers and use only the allocated space.
You may color your answers blue for easy grading.

Objective: To deepend the understanding of time and space complexity in search algorithms and deciding on suitable algorithms for a given problem.

Type your answers, but you can draw any illustrations by hand (if so you can send the scanned document).

1) 30pts –**Answer the following using the general Tree Search algorithm** (remove front node from the fringe/ queue – goal test – expand).

Reminder: You can use the following equality for compactness:

$1 + b + b^2 + \dots b^d = (b^{d+1}-1)/(b-1)$

a) 15pt - How many nodes are ***visited*** (chosen from the queue, goal tested and expanded) in the worst case using Breadth-First search, when the solution is at depth d, and the branching factor is b, and the depth of the maximum branch is m?

Give a formula.

$1 + b + b^2 + \dots b^d = (b^{d+1}-1)/(b-1)$

…………………………………………………………………………………………..

b) 15pt- How many nodes are ***generated*** (added to the queue as a result of expanding the parent) in the worst case using Breadth-First search, when the solution is at depth d, and the branching factor is b, and the depth of the maximum branch is m?

$b+ b^2 +\dots.+b^{(d+1)-b} =[(b^{d+2}-1)/(b-1)]-1$

…………………………………………………………………………………………..

2) 45pt – You are given the problem of finding whether 6-degrees of separation holds between a particular 2 people in the world. E.g. given two people – say you and your favorite celebrity - the software should decide whether they are connected in at most 6 friendship edges (e.g. you-f1-f2-f3-f4-f5-celebrity).

Let`s assume you have the list of all friendships for all people in the world and that everyone has exactly b=100 friends and that there are 6 billion people in the world.

a) **18pts)** State **whether the following algorithms are <u>complete</u>** (if there is a up to 6-degree path, does it find it?) **and <u>optimal</u>** (defined here as 'does it find the shortest path connecting two people') **for this problem**.

b) **12pts) If an algorithm is BOTH complete AND optimal, comment on its time and space complexity with a one line summary about its suitability (e.g. "will take too much time/space: O(b^d)").** If an algorithm would take too much time or space to be feasible, indicate as such; if it is suitable but is an overkill, you should indicate that also.

Algorithm	**Complete (answer as Yes or No)**	**Optimal (answer as Yes or No)**	**Feasibility (add a one line comment)**
Breadth first search	YES	YES (if step costs are uniform)	It has time and space complexity O(b^(d+1) and memory requirements are huge problem in bfs because every degree network of first given people will be generated.
Depth first search without repeated state checking	NO	NO	
Depth first search with repeated state checking	YES	NO	
Depth limited search DFS with a depth limit of ……6……	YES	YES	If there is up to 6 degree path it will find it and it's optimal because it's memory requirements are modest. It only takes O(bl) memory.Time requirement is O(b^6).
Iterative deepening DFS	YES	YES	It has time complexity O(b^d) and space complexity O(bd). It may be wasteful because stares are generated multiple times.It's preferred when the search space is large and depth of the solution is unknown.

Bidirectional search	YES	YES	Time and space complexity is $O(b^{b/2})$.It should be used only when the goal is well defined

c) **15pts) Which blind search algorithm** (among the ones listed above**) would be best for this problem? Explain your answer.** Consider space, time complexities and completeness and optimality.

If two algorithms are the same or similar, you may choose the one which is easier to implement or state that they are both as good / suitable.

Bidirectional search should be choose to implement because the time and space requirements are less than other solutions which is $O(b^{b/2})$ and our starting point and the goal points are explicitly defined and the search operators are reversible. It will run bds simultaneously from the start and goal nodes. Bidirectional search is useful branching factor is large and the diameter of the graph is small and 6 degree separation problem is suitable for this.