**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 + b2 + … bd = (bd+1-1)/(b-1)

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 + b2 + … bd = (bd+1-1)/(b-1) = O(bd )

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 + b2 + b3 + …… + bd + b(d+1) - b  = (bd+2-b -1)/(b-1) = O(bd )

**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.**

1. **18pts)** State **whether the following algorithms are complete** (if there is a up to 6-degree path, does it find it?) **and optimal** (defined here as ‘does it find the shortest path connecting two people’) **for this problem**.
2. **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 otherwise **No**) | The space complexity would be very high:  O(bd+1). The time complexity will also be a problem: O(bd ). (b = 100 and d = 6) |
| 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** | **No** |  |
| Iterative deepening DFS | **Yes** | **Yes (**when the path cost is a nondecreasing function of the depth of the node) | The time complexity would be high:  O(bd) and the space complexity would be O(bd) which is feasible compared with the BFS algorithm(b = 100 and d = 6). |
| Bidirectional search | **Yes**  (if both directions use breadth first search) | **Yes**  if both directions use breadth first search and steps costs are identical) | The time and space complexity would be: O(bd/2) (b = 100 and d = 6) which is feasible for our case but performs worse compared with iterative deepening search in terms of memory aspect. |

1. **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.

Both breadth first search, iterative deepening search and bidirectional search is complete and optimal in our case. But if we consider the time complexity most, then bidirectional search would give us a better result by far since it reduces the time complexity a lot. Also, since both initial and goal states are unique and well-defined in our problem, bidirectional search is appropriate for our problem. Hence bidirectional search is the best for the given problem. On the other hand, if we have a very limited memory and time is not much important, then iterative deepening search would be better for us.