

CS 2710/ISSP 2160

Homework 1 Answer key

1 Formulating search problems (10 pts)

Grading Criteria:

Each of the subproblems (a) and (b) was graded for 10 pts, then scaled down to 5. In both subproblems, each of the five pieces of information: state, initial state, successor function, goal test, and cost function were worth 2 points. Full credit was awarded for each piece if it included some answer similar to the solutions, or another variant that represented the problem in an accurate way, which was precise enough to be implemented. One point was awarded for solutions that included a good deal of the solution but were not quite right, or at least a good attempt. Zero points were awarded for solutions, which were too vague or completely wrong.

- (a) **States:** An unordered list of un-stacked items which can include yourself, the fridge and the chair. An ordered list of stacked items where items earlier on the list are stacked under items later on the list (i.e., a 2-tuple containing `<un-stacked list, stacked list>`)

Initial state: Initially `<{you, chair, fridge}, {}>`.

Successor function: The actions that can be performed are moving any item from the un-stacked list to the stacked list, or moving the last item from the stacked list to the un-stacked list. That is:

```
Successor(<{you, fridge, chair}, {}>) = {
    <stack(you), <{fridge, chair}, {you}>>,
    <stack(fridge), <{you, chair}, {fridge}>>,
    <stack(chair), <{you, fridge}, {chair}>> }
Successor(<{you, fridge}, {chair}>) = {<stack(fridge), <{you}, {chair, fridge}>>, ...}
```

All successors need not be shown in the solution, just enough to demonstrate a good understanding of what the successor function is.

Goal test: Any state where you have `<{}, {x, x, you}>` where x can be any item, or alternatively (a more robust goal test) any state such that stacked set has you on the right and summing the heights of the items in the list leads to a height ≥ 9 feet.

Cost function: The following cost functions were acceptable

- The number of moves you make.
- Some form of cost using hypothetical weights of the items and/or the distance they were moved and/or the height they were lifted.

(b) **States:** The set of actors guessed so far and the last actor guessed, i.e. the pair $\langle X', x \rangle$, where $X' \subseteq X$ is the set of guesses, and x is the last actor.

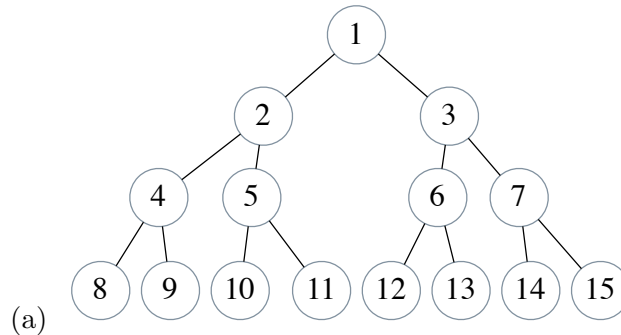
Initial state: There have been no guesses, and the first player may begin by guessing any actor, i.e. $\{\langle \emptyset, x_i \rangle \mid x_i \in X\}$. **Successor function:** Add an actor y to the current set of guesses such that y is a co-star of x in some movie, i.e. $\text{successor}(X, x) = \{\langle \text{add}(y), (X \cup \{y\}, y) \rangle \mid y \text{ co-stars with } x\}$.

Goal test: Any state with no successor state. That is $\langle X', x \rangle$ such that there are no actors y co-starring with x that have not already been guessed.

Cost function: The number of guesses so far, $|X'|$.

2 Blind search (10 pts)

(R&N Problem 3.8) Grading criteria: Parts (a) and (b) are assigned 4 and 6 pts, respectively. Part (b) is graded independently of correctness of (a). In part (b) each search technique is assigned 2 pts. Since solutions cannot be partially correct in this problem, no partial credit is given.



(b) Goal state: 11

Breadth First Search: 1, 2, 3, 4, 5, ..., 11.

Depth Limited Search (limit = 3): 1, 2, 4, 8, 9, 5, 10, 11.

Iterative Deepening Search: Each line is an ordered sequence of nodes visited in an iteration.

Iteration 0: 1;

Iteration 1: 1, 2, 3;

Iteration 2: 1, 2, 4, 5, 3, 6, 7;

Iteration 3: 1, 2, 4, 8, 9, 5, 10, 11.

3 Heuristic search (10 pts)

Grading criteria: each part is assigned 3 pts except for part (b) (4 pts). In each part, subpart (i) is assigned 2 pts, leaving 1 pt for (ii) (2 pts in case of part (b)). No partial credit.

(a) Uniform cost search

(i) Each line is a list of nodes generated due to expanding the leftmost node. The lines are listed in the order of node expansion:

S: A(2), D(3)

A: B(6) (Note that S is pruned as a child of A since the cost to S through A is greater than the cost of best known route to S (which has cost zero as initial state))

D: E(6). S and A are pruned.

B: F(9), C(10). A and E pruned.

E: B, D, and F are pruned.

F: G(15). B and E are pruned.

C: G(11). B is pruned.

G: Goal reached.

(ii) Solution path: S,A,B,C,G. It is the shortest (cost is 11).

(b) Greedy Best-First Search

(i) Solution layout as in part (a)

S: A(7), D(5)

D: S(10), A(7), E(4)

E: D(5), B(3), F(2)

F: E(4), B(3), G(0).

G: Goal reached.

(ii) Solution path: S,D,E,F,G. It has cost 15. (not optimal)

(c) Recursive Best-First Search

(i) Solution layout as in part (a). Note that unlike greedy best-first search, here each node is labeled with its f -value

S: A(9), D(8). New f -limit for S's subtree is $f(A) = 9$

D: S and A are pruned. E(10) fails since f -limit is $f(A) = 9$. New f -limit for S and D's subtrees is $f(E) = 10$

A: S and D are pruned. B(8): f -limit remains 10 since the second best of A's subtree is ∞ (no second best)

B: A and E are pruned. C(11), F(11) fails due to f -limit 10. New f -limit is $f(C) = 11$

D: S and A are pruned. E(10). f -limit remains 11.

E: B, D and F pruned, failure due to empty set of successors.

A: B(8): f -limit remains 11

B: A and E are pruned. C(11), F(11). f -limit remains 11.

C: G(11). f -limit remains 11.

G: Goal reached.

(ii) Solution path: S,A,B,C,G. It has cost 11. (optimal)

4 A* search (10 pts)

Grading criteria: Each part is assigned 5 pts.

- (a) The given heuristic is admissible because those white tiles would need to move to the left of the leftmost black tile to have a goal state. The cost of doing this at the least is the #white tiles for the black tile to hop over.
- (b) List 10 generated nodes. Two pts are taken out if expanded nodes are not shown with the generated nodes.

5 Empirical evaluation of heuristic search (60 pts)

Grading criteria: The problem is graded as three parts: (a) demonstration of implementation correctness, (b) evaluation of heuristics (Manhattan and M&P heuristics) using A*, and (c) evaluation of heuristics using IDS. In each part, correct implementation is worth 10 pts. Analysis (documentation in case of part (a)) is assigned the remaining 10 pts.