Daniel Willard
471 Intro to AI
Assignment Written 1
Professor Thanh Nguyen

Q1) Word Ladder

1) **Formulate the problem of solving word ladders as a search problem.**
    - **a.** Starting State: The Starting Word form example "Cold"
    - **b.** State Space: the number of letters in the word, the given letters in the language dictionary (to create another word of the same size that is in the dictionary given)
    - **c.** Goal State: The target Word form example "Warm"
    - **d.** Successor Functions: Generate all valid words that can be reached by changing a single letter in the current word to form a new word. This will be constrained by a given dictionary, Position in the word will need to be tracked and the replaced letter will need to be track in a some form.
    - **e.** Successor Functions Actions: the swapping of a single letter in a single position in the current word to form a new word. from example cold ->cord where 'l'->'r' in position 2 if initial position is 0.
    - **f.** Successor Functions Cost: In this case all swaps are considered cost 1, thus constant and can be omit.
2) State Space Graph for converting "bee" to "dad" using only the first seven letters (a-g):
    - **a.** bee

```
   /  \        \     \
 bed   beg  bae  fee
   |     |
 bad   bag
   \    /
    dad
```

3) Which part of the search tree is expanded by DFS? (this feels unclear)
    - a. DFS is a LIFO algorithm of expanding nodes. This means it will explore one branch of the search tree to the end of the branch before iterateing till there is no otherr branches.
    - b. ie. bee->bed->bad->dad [goal meet would not iterate, but if it did it would expand] beg->bag->dad.
        - i. node fringe
        - ii. bee bee
        - iii. bed bee->bed
        - iv. bad bee->beg
        - v. dad bee->bed->bad
        - vi. bee->bed->bad->dad
    - c. so in DFS the part of search tree that is expanded is the deepest unexplored branch, moving deeper until a solution is found or all possible paths are exaughsted.

referenced: https://www.geeksforgeeks.org/search-algorithms-in-ai/

Q2) Graph Search

1) **In what order are states expanded by UCS? What path would UCS return?**
   a. UCS states are expanded by the accumulated path cost form the start node to the goal node with alphabetically order breaking ties.

queue:
0:(S,0)
S0:(SA,1)(SB,2)(SC,2)
A1:(SB,2)(SC,2)(SAD,4)(SAF,3)
B2:(SC,2)(SAD,4)(SAF,3)(SBD,5)(SBF,6)
C2:(SAD,4)(SAF,3)(SBD,5)(SBF,6)(SCD,3)(SCF,4)
F3:(SAD,4)(SBD,5)(SBF,6)(SCD,3)(SCF,4)/(SAFG,6)
D3:(SAD,4)(SBD,5)(SBF,6)(SCF,4)/(SAFG,6)(SCDG,5)
D4:(SBD,5)(SBF,6)(SCF,4)(SAFG,6)(SCDG,5)(SADG,6)
F4:(SBD,5)(SBF,6)(SAFG,6)(SCDG,5)(SADG,6)(SCFG,7)
Goal: SCDG,5
Step 1: Expand S (cost: 0).
Step 2: Expand A (cost: 1).
Step 3: Expand B (cost: 2).
Step 4: Expand C (cost: 2).
Step 5: Expand F (cost: 3).
Step 6: Expand D (cost: 3).
Step 7: Expand G (cost: 5) - Goal reached.

Expansion Order:S,A,B,C,D,F,G
Returned Path: S(0)->C(2)->D(3)->G(5) completed found goal in cost 5 optimal

2) **In what order are states expanded by Greedy? What path would Greedy return?**
   a. Greedy states are expanded based on their heuristic value(h-value) it will always choose the node with the lowest h value with alphabetical order used to break ties.

Step 0: Initialize with S (cost: 0).
Step 1: Expand S (h-value: 5).
Step 2: Expand A (h-value: 3).
Step 3: Expand F (h-value: 1).
Step 4: Expand G (h-value: 0) - Goal reached.

Expansion Order: S,A,F,G
Returned Path: S(5)->A(3)->F(1)->G(0) completed found goal

3) **In what order are states expanded by A*?What path would A* return?**
   a. A* states are expanded based on the combine value of the fallowing: actual cost so far to reach node n (g-value), and heuristic value(h-value).
   b. Then expands on the path that are already less expensive using the function f(n) = g(n)+h(n). with alphabetical order breaking ties

Step 0: Initialize with S (cost: 0, f-value: h(S) = 5 + 0 = 5).
Step 1: Expand S (cost: 0, f-value: 5).
Step 2: Expand A (cost: 1, f-value: 1 + h(A) = 1 + 3 = 4). SA
Step 3: Expand F (cost: 3, f-value: 3 + h(F) = 3 + 1 = 4). SAF
Step 4: Expand D (cost: 4, f-value: 4 + h(D) = 4 + 2 = 6). SAD
Step 5: Expand B (cost: 2, f-value: 2 + h(B) = 2 + 4 = 6). SC
Step 6: Expand C (cost: 2, f-value: 2 + h(C) = 2 + 4 = 6). SB
Step 7: Expand G (cost: 6, f-value: 6 + h(G) = 6 + 0 = 6) SAFG - Goal reached.

Expansion Order:S,A,F,D,B,C,G
Returned Path: S(0)(5)->A(1)(3)->F(3)(1)->G(6)(0)

Q2.2) Admissibility/Consistency.

1) **Is the above heuristic function h admissible? Explain your answer.**
   a. NO, for a heuristic function to be admissible it must never overestimates the true cost to reach the goal from any given node. In other words for all 'n' nodes the heuristic value 'h(n)' must be less then or equal to the actual cost to reach goal 'goal(n) form node 'n'

Testing each node:

S->G: h(S) = 5 <= 5(S->C->D->G)
A->G: h(A) = 3 <= 5(A->D->G)
B->G: h(B) = 4 <= 5(B->D->G)
C->G: h(C) = 4 <= 3(C->D->G) *fails
D->G: h(D) = 2 <= 2(D->G)
F->G: h(F) = 1 <= 3(F->G)
G->G: trivial case h(G) = 0 <= 0.

Therefore, the heuristic function is not admissible because it overestimates the cost to reach the goal from node C.

2) **For which TWO states (S, A, B, C, D, F, or G) could you change the heuristic values to make everything consistent? What range of values are possible to make this correction?**

State of C must be changed to a value of [3] anything else breaks consistancy.
For S -> C:
h(S) = 5
c(S, C) = 2
h(C) = 4
$5 \leq 2 + 3$

For C -> D:
h(C) = 4
c(C, D) = 1
h(D) = 2
$3 \leq 1 + 2$

For C -> D:
h(C) = 4
c(C, D) = 1
h(D) = 2
$3 \leq 1 + 2$

State of A should be changed [4,5] depending on F

For S -> A:
h(S) = 5
c(S, A) = 1
h(A) = 3
$5 \leq 1 + 4$ (Consistent)

but that would force State at F to be change as well

For A -> F:
h(A) = 3
c(A, F) = 2
h(F) = 1
$3 \leq 2 + 2$

State of F should be changed to [2,3]

Q3) Hive Minds: Lonely Bug

1) **Provide a minimal correct state space representation? What is the state-space size?**
    - **a.** The state space representation for the insect navigating in a rectangular maze-like environment can be represented as follows:
        - **i.** State Space: current postion (x,y) of the insect in the maze
        - **ii.** Initial State: Starting Postion (x,y)
        - **iii.** Goal state: Target Location (x,y)
    - **b.** State Space size is M*N for very possible box postion in the maze.

2) **Provide two heuristics which are admissible.**
    - **a.** Two admissible heuristics for this problem could be:
        - **i.** Manhattan Distance Heuristic:
            - **1.** It is admissible because it underestimates the true cost to reach the goal, as the insect can only move in four directions (up, down, left, and right)
            - **2.** The Manhattan distance heuristic is given by: h_manhattan = |x_current - x_target| + |y_current - y_target|
    - **b.** Euclidean Distance Heuristic:
        - **i.** It is admissible because it also underestimates the true cost to reach the goal, as it assumes the insect can move directly towards the target without any obstacles.
        - **ii.** The Euclidean distance heuristic is given by: h_euclidean = sqrt((x_current - x_target)^2 + (y_current - y_target)^2)