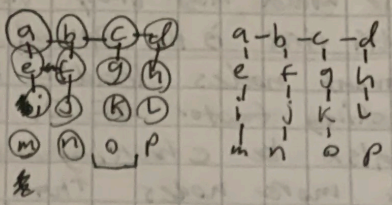
Sam Hopkins
CS 470
Hw 6

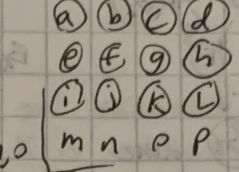
1.



a) List of states expanded: a, b, e, c, f, i, d, g, j, m, h, K, n, L, O

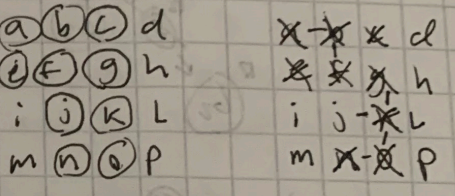
Path: a, b, c, g, K, O

b) List: a, b, c, d, h, g, f, e, i, j, K, L, m, O



Path: a, b, c, d, h, g, f, e, i, j, K, L, m, n, O

c) List: a, O, b, K, e, n, c, p, f, g



Path: a, b, e, g, K, O

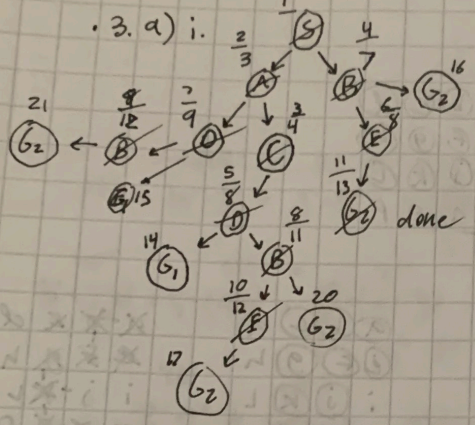
2. a) i. Each state is a person in Spain, or the set of all people born in Spain from Segio until now.
ii. Sergio will be the initial state
iii. Available actions include finding either the parents of a person or children, with equal cost. finding Parent won't help us because we are starting with Sergio, so finding the children would improve our state the most.
iv. Our goal state is Alberto

b) no, you also could start with Alberto and have a goal state being Sergio. Then you would simply need to travel up through the Parent() states until you reach Sergio

c) Starting with Alberto. Because population has increased so much, we know that there will be a lot more options expanding children than expanding parents. Each child will only have 2 parents, but each parent could have n children, which increases our branching factor quite a bit.

d) Depth first would not be the best as because it doesn't mark nodes as visited, it would most likely visit too many nodes to be efficient. Breadth first is the optimal, because we would only need to check Parent nodes leading from Alberto to Sergio, which has a branching factor of 2. Bi-directional is worse because we would also be checking children of sergio which will generate way more nodes than simly checking parents. (would have a branching factor of n)

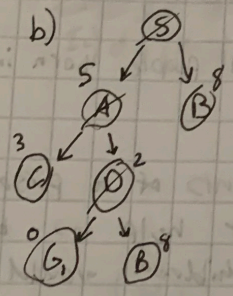
3. a) i.




~~[illegible]~~
Path: S, B, E, $G_2$

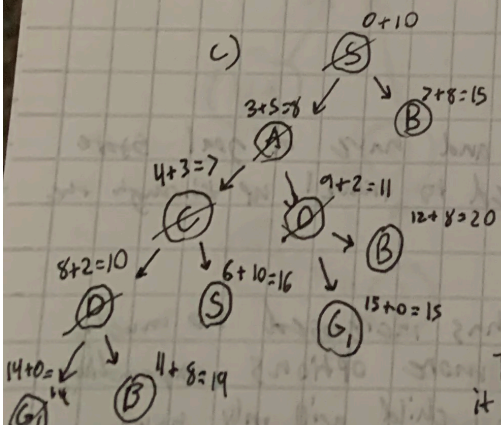
Expanded: S, A, C, S, B, D, E, A, D, C, B, B, E, S, B, $G_2$

b)




Expanded: S, A, D, $G_1$
Path: S, A, D, $G_1$

c)




This makes it so the only optimal value is found by UCS

Expanded: S, A, C, D, D, $G_1$
Path: S, A, C, D, $G_1$

The heuristic value is off, so algorithms that use it get a different answer than we want.

d) h is admissable if $h(n) \leq C^*(n)$
It is not admissable because h(n) (~~[illegible]~~ cost to goal state of A*) > $h^*(n)$, or the minimal cost to get to a goal state. $h^*(n) = 13$, while $h(n) = 14$. Because A* did not find the optimal solution, h is not admissable.

4a) Given $f(N') \leq (1+\epsilon)\left[\min_{M \in \text{fringe}} \{f(M)\}\right]$, $h_A(N)$ is admissable, costs are greater than some finite $\epsilon$

> Node $g'$ (suboptimal goal node in fringe) will not be expanded because $K^{(\text{optimal})}$

1. f-value decides if expanded
2. $f(g') = g(g') + h(g')$
3. We know $g'$ is a goal node (suboptimal) $g(g') > C^*$ and $h(g') = 0$
4. $f(g') = g(g') + h(g') = g(g') > C^*$
5. f-value of $K$: $f(K) = g(K) + h(K)$ $h(K) = 0$ $g(K) = C^* \rightarrow f(K) \leq C^*$

So will $g'$ be expanded? With new heuristic:

1. $f(g') = (g(g') + h(g'))(1+\epsilon)$
2. $f(K) = (f(K))(1+\epsilon)$
3. If $\epsilon > 0$, then $g'$ could get expanded.
4. $f(g') \leq (1+\epsilon)\left[\{f(K)\}\right]$, because K is minimum goal node in the fringe.

ex. If $\epsilon = 1$ and $f(g') = 2$ $f(K) = 1$ then $2 \leq (1+1)(1)$ is true, So in such a case $g'$ could be expanded, giving the $A^*$ algorithm a sub optimal goal node, and $h_N$ is not admissible

The cost of the solution will be $f(K) \leq C^*(1+\epsilon)$ where $f(K)$ is the returned solution cost. I $\epsilon > 0$, then it could return a cost that is suboptimal.

b) Because greedy search algorithms are significantly faster than $A^*$, $A_\epsilon^*$ allows for a type of greedy algorithm as well, because as long as a node is $(1+\epsilon)$ times the minimal node than it can be expanded. This can give us a greedy solution with limits. With $\epsilon$ being larger, than more nodes can be expanded, so more greedy solutions will be found. If $\epsilon$ is small, then we get more of an $A^*$ solution.

5. a) i. Each state would be the position of each car on the grid. Each state transition would be the different states the cars could be in given the previous state.

ii. The initial state is shown in the picture, with each vehicle populating the first row (1, i)

iii. Goal state is each car in the $(n, n-i+1)$ position in the grid, as shown in the example

b) We have $n^2$ cells, and we are looking at all the ways to put $n$ cars in $n^2$ cells, so we end up with a state space of $\underline{n^{2n}}$ (iii)

c) $5^n$, each car $n$ has 5 different moves it can do (L, R, U, d, stay), so each state is a combination of those 5 for $n$ cars, so $5^n$

d) Because we have squares, we would use the manhattan distance to its goal state, so $(r_i, c_i) \rightarrow (n, n-i+1)$ $h_i = |n - r_i| + |(n-i+1) - c_i|$, one for each grid movement.

e)

i. yes, for it to be admissible, h at goal = 0, and if 0 is in $\min\{h_1, \ldots h_n\}$ then it would be returned, so it is admissable, as that would be the optimal solution

ii. No because at the goal state we would not get the optimal solution. This will just say how many states we generate at any given n, but is not an admissible heuristic.