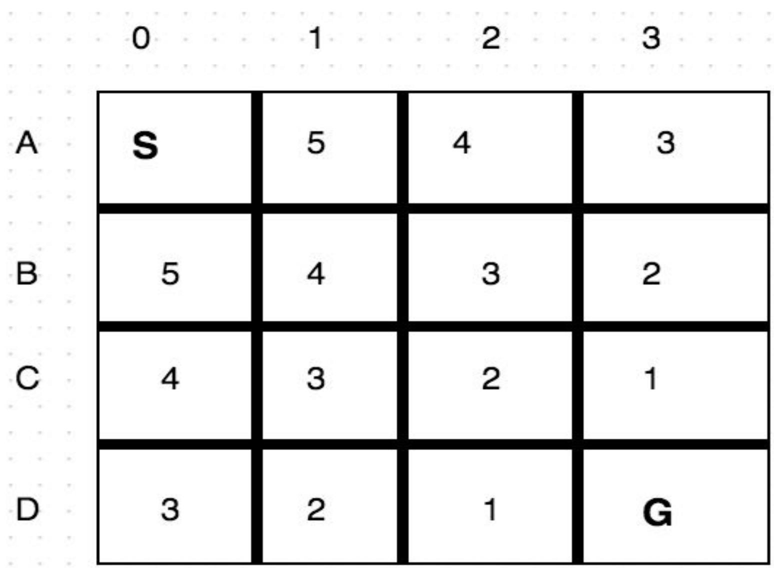
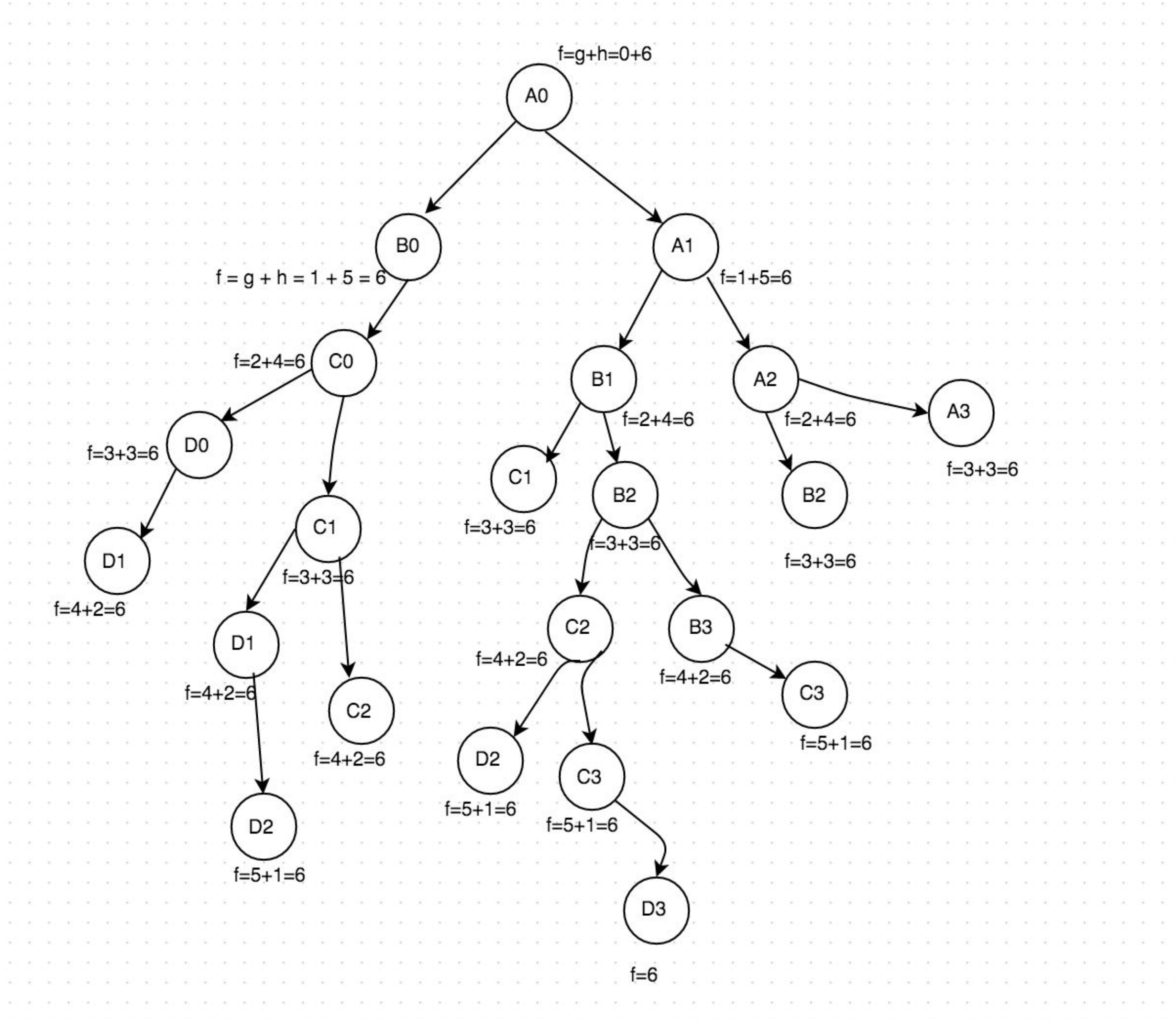
Project 1 (Part A)

Given that the both versions always expand the unexpanded fringe node with the smallest f-value, the heuristics for both of these methods are admissible and consistent, and hence, they are optimal. One of the properties of consistency claims that f(n) ≤ f(n'). In this problem, given that C is the cost of optimal solution path, then both versions of A* search expand all nodes with f(n) < C. However, since we are more concerned about the number of expanded nodes, the distinction between both versions can be clearly seen by displaying the tree representations of both searches:

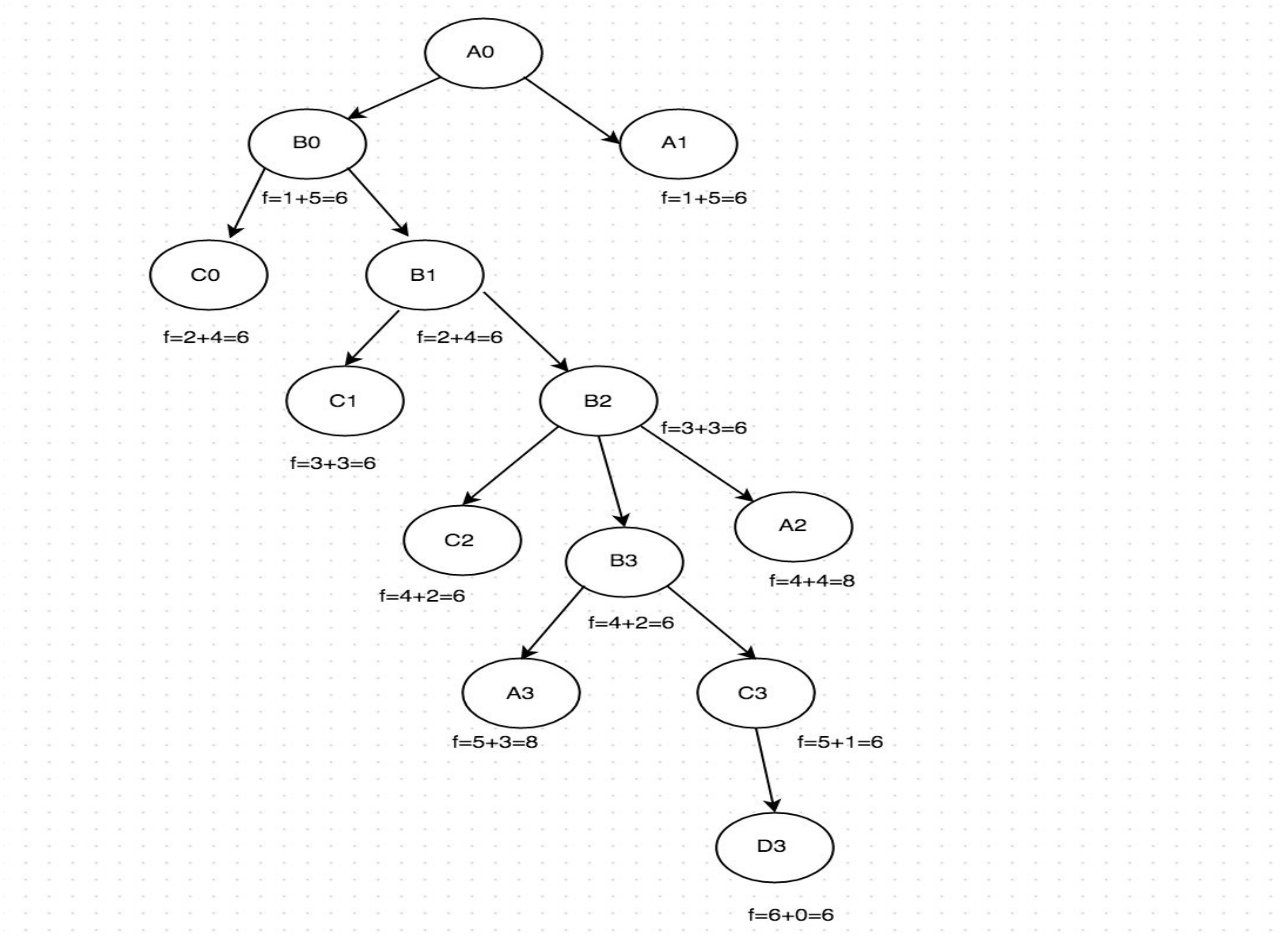
Let's take a 4x4 grid that has the following structure:



If we do the A* search with the version of the smallest g, we will get the following tree:



As we can see in the graph above, A* search with the smaller g expands a lot of nodes. Let's see what happens with the A* search of the larger g preference.



By looking at the tree above, we can clearly see that the A* search with the larger g preference expands much less nodes. Why? If we analyze the formula given in class [0 <= h(s) <= c(s, s') + h(s')], the optimal cost is getting further away from optimality. Because the smallest g has more expansions. In other words, if we choose the biggest h (and h=0 at the goal state), and choosing the biggest h means choosing the furthest point from the goal. Hence, the second version of the A* search expands less nodes due to smaller g. Since in both versions we keep the same standard of optimality, the ratio between both versions of A* search doesn't change and is therefore true in all cases.