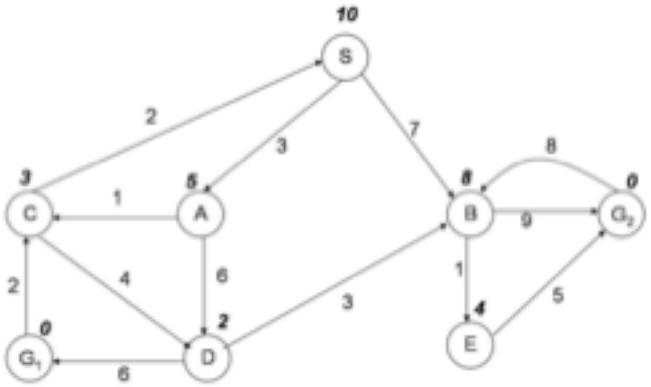
Consider the following state graph:



The states consist of *S, A, B, C, D, E, G_1*, and *G_2*. *S* is in the initial state and *G_1* and *G_2* are the goal states. The possible actions between states are indicated by the arrows. So, the successor function for state *S* returns {*A, B*}; for *A* it returns {*C, D*} and so forth. The number labeling each arc is the actual cost of the action (i.e. *S* to *A* is 3). The number above each state is the value of the heuristic function *h* at that state. For example, the value of *h* at state C is 3.

Build the search tree using the A* algorithm as described in class. Run the algorithm first for *G_1*.Then repeat the algorithm for node *G_2*. (You will have 2 different search trees.) The algorithm does not check if a state is a revisited one or not, hence there may be several nodes representing the same state in your tree. The states produced by the successor function are always ordered (i.e. returned) in alphabetic order. The algorithm only terminates when a goal node is removed from the fringe.

Lastly, consider the question if the heuristic function *h* defined in the above graph is admissible?