HÁSKÓLINN Í REYKJAVÍK REYKJAVIK UNIVERSITY	T-622-ARTI, Introduction to Artificial Intelligence Spring 2011 **Problem Set 1** Worth 5% of final course grade Due by 23:59 on Thursday the 17th of February

Student Name: ____

1. (25%) Formulating Search Problems

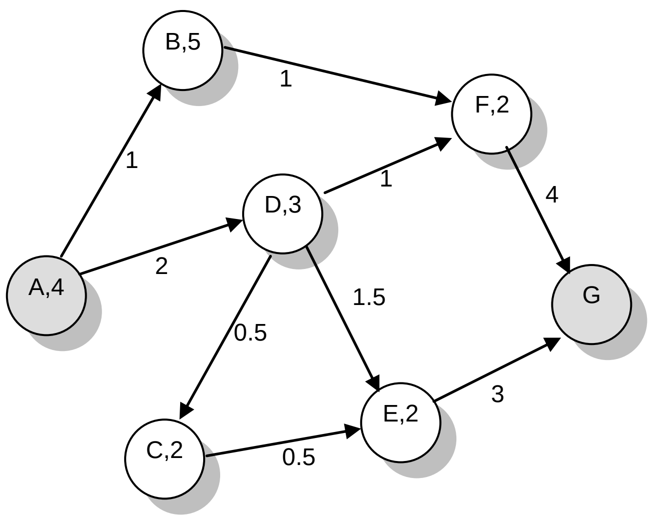
Formulate the following search problems by giving an *initial state, goal test, actions description, transition model (successor function)* and *cost function*. Choose a formulation that is precise enough to be implemented at your chosen level of abstraction.

a. Suppose there are two friends living in different cities on a map, such as the Romania map shown in Figure 3.2 of the textbook. On every turn, we can move each friend simultaneously to a neighboring city on the map. The amount of time needed to move from city i to neighbor j is equal to the road distance d(i; j) between the cities, but on each turn the friend that arrives at their destination first must wait until the other one also arrives at their destination (and calls the first on his/her cell phone) before the next turn can begin. We want the two friends to meet as quickly as possible.

b. Suppose that you are looking at a map of Europe showing each of its states and that we are given the task of coloring each region in such a way that no adjacent countries have the same color. We are given a large set of colors but we want to use the fewest possible.

2. (25%) Performing an A* Search

The following state-space is given, with each state showing its *state id* and the value of its heuristic estimate (*h(s)*) of how far it is from a goal state. Edges indicate the successor function and their actions are labeled with their cost. The start state is *A* and G is the goal state.



a) Are the heuristics in this problem *admissible*? How do you see that?

b) Use the A* algorithm to find the optimal path to a goal state. Tie-breaking of equal *f*-values on the *Open list (frontier)* is done alphabetically (lower values first). A search node consists of the *state id*, *g-value*, *h-value*, and the *parent state id*, e.g. *[A, 0, 4, -]*. The table below shows the open and closed lists when tracing an execution of the algorithm. Continue the trace until you have identified a path to a goal state. Show the contents of both the open and closed list after each (expansion) step.

Open List (frontier)	**Closed List**
[A, 0, 4, -]	
[B, 1, 5, A], [D, 2, 3, A]	[A, 0, 4, -]

3. (25%) Inventing Heuristics

Suppose you're trying to solve the following puzzle. The puzzle involves numbers from 100 to 999.
You're given two numbers called *S* and *G*. You're also given a set of numbers called *bad*. A move consists of transforming one number into another by adding 1 to one of its digits or subtracting 1 from one of its digits; for instance, a move can take you from 678 to 679; or from 234 to 134. Moves are subject to the following constraints:

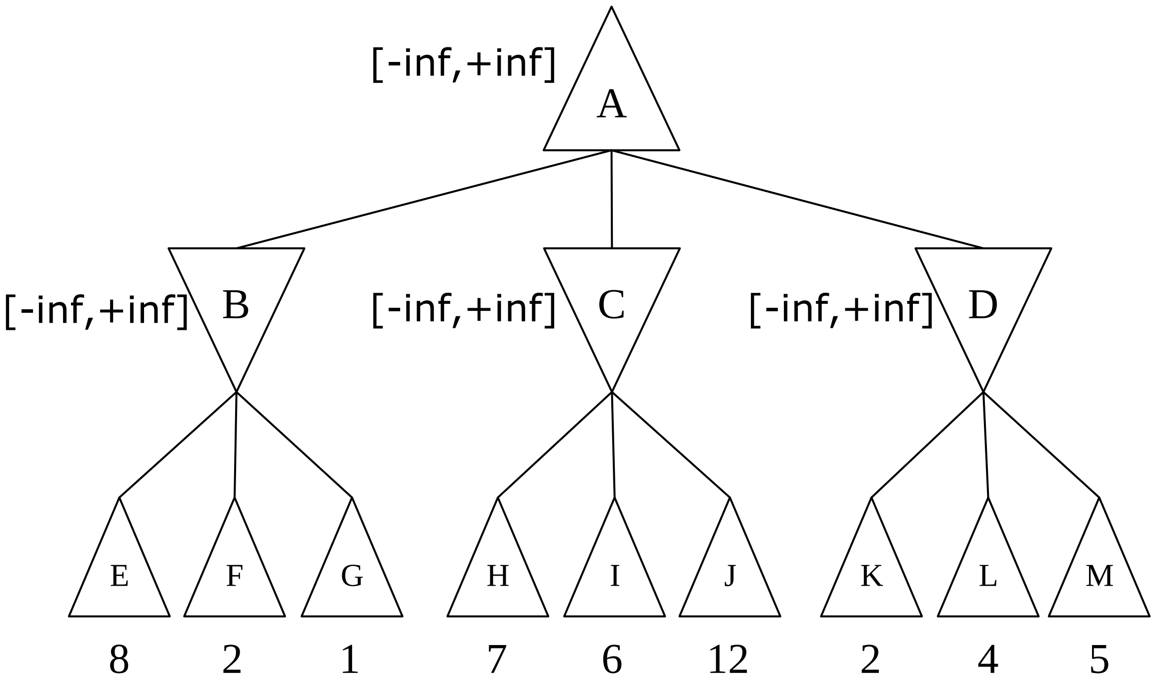
- You cannot add to the digit 9 or subtract from the digit 0. That is to say, no "carries" are allowed and the digits must remain in the range from 0 to 9.
- You cannot make a move which transforms your current number into one of the numbers in the set *bad*.
- You cannot change the same digit twice in two successive moves.

Since the numbers have only 3 digits, there are at most 6 possible moves at the start. And since all moves except the first are preceded by another move which uses one of the digits, after the start there are at most 4 possible moves per turn. You solve the puzzle by getting from S to G in the fewest possible moves.

Imagine that you need to solve this puzzle using an A* search (no need to actually solve it).

- **a.** Briefly list the information needed in the state description (not the node description) in order to apply A* to this problem.
- **b.** What would be a good heuristic function to choose? Provide arguments. Show that your function is *admissible*. Is it also *consistent*?

4. (25%) Minimax algorithm with Alpha-Beta Pruning



The picture above shows a two-ply game tree. The △ nodes are "MAX nodes", in which it is MAX'S turn to move, and the ▽ nodes are "MIN nodes."
The [-inf, +inf] are place holders for the alpha-beta bookkeeping. Alpha is initialized to negative infinity (-inf). Beta is initialized to positive infinity (+ inf).
If we use the ***minimax*** algorithm to traverse the tree above, in a left-to-right manner, the nodes are visited in the following order: *A, B, E, F, G, C, H, I, J, D, K, L, M.*

a. What is the **game tree value** at the root?
b. Fill in the Alpha and Beta values and show the pruned branches applying the ***minimax*** algotithm with a***lpha-beta*** pruning (use the tree below):