Homework #2 (30p)
Due: Tue. Sep. 30 at the start of the class

Note #1: Please hand print your answers. If we cannot read them we will assume that they are wrong.
Note #2: A programming homework will be assigned soon. Start this one as soon as possible.

1. (5p) The 8-puzzle consists of eight numbered, movable tiles set in a 3×3 frame. One cell of the frame is always empty thus making it possible to move an adjacent numbered tile into the empty cell—or, we could say, to move the empty cell. Two configurations of tiles are given. Consider the problem of changing the start configuration into the goal configuration.

1	2	3
4		8
5	6	7

start

2	4	3
1	6	8
5		7

goal

Set up a state-space search formulation for this puzzle:

a) (5%) Specify the form of state descriptions, the starting state, and the goal state for this problem.

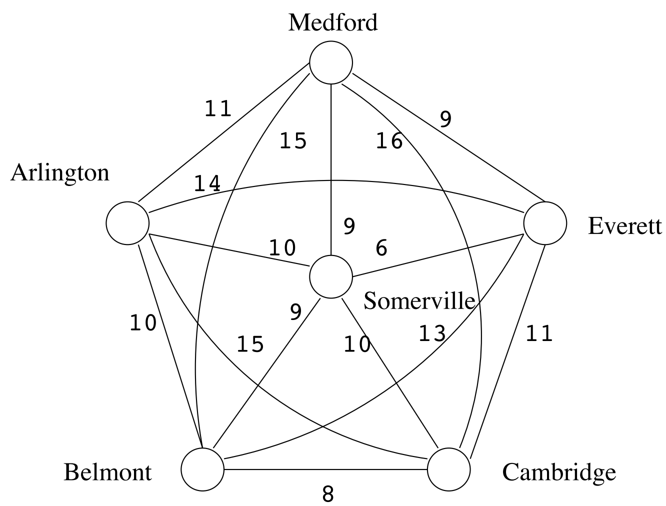
b) (15%) Name the operators on states and describe what each operator does to a state description.

c) (30%) Propose two heuristic function $\hat{h}$ for solving this problem. Which heuristic function is better? Justify your argument.

d) (40%) Show all nodes expanded in the first two steps of the A^* algorithm for both heuristics. Show the associated costs.

e) (10%) What condition is necessary to make the solution path obtained by A^* optimal? Give a formal argument.

2. (5p) The Traveling Salesman Problem (TSP) is a challenge to the salesman who wants to visit every location exactly once and return home, as quickly as possible. Each location can be reached from every other location, and for each pair of locations, there is metric that defines the time between them. Given the following graph



your task is to use hill-climbing to find the local or global minimum for this problem. Start with the loop *(Medford Belmont Everett Arlington Cambridge Somerville Medford)* [or *(M B E A C S M)* for short]. Once you find the local minimum you can stop. Note that at each step the method should identify a pair of edges whose endpoints can be switched to improve the loop. [Finally, strictly speaking this is not climbing, but descent.]

3. (5p) Use different techniques to find feasible solutions for 4×4 sudoku puzzles. In a 4×4 sudoku puzzle numbers $1-4$ have to appear exactly once in each row, column and a 2×2 cell (the grid has 4 such cells: top left cell contains grid elements $(1,1)$, $(1,2)$, $(2,1)$, and $(2,2)$; top right cell contains grid elements $(1,3)$, $(1,4)$, $(2,3)$, and $(2,4)$; bottom left and bottom right cells are defined similarly). The start puzzle has numbers $3,1,2,4$ in the first column and all other grid-cells are empty. Solve this problem using the following techniques:

   a. Backtracking search.
   b. Forward checking.

4. (5p) Show how min-conflicts algorithm can be used to solve 5-queens placement problem (place 5-queens on a 5×5 board so that there is at most one queen in each row, column, or a diagonal of the board). To initialize the problem place all five queens in the first row of the board. You need to cycle through all queens at most twice. If you do not get a solution leave it at that.

5. (5p) A simple version of the *Nim* game is played as follows: Two players alternate in removing stones from a single pile initially containing 7 stones. The player who picks up the last stone wins. The following rules must be observed:

   1. Each player can pick one, two, or three stones.
   2. At every turn the players have to pick a number of stones different from the number picked by their opponent on the preceding move.
   3. Players must take stones if they can.

Show, by drawing the game tree, whether or not the first player can always win.

6. (5p) Apply minimax with alpha-beta pruning on the following graph. Show the backed up values, the places where alpha-beta prunes (the cuts) and the reason for each cut. Reasons can be simple (i.e., "8 < 15" or "12 > 6").