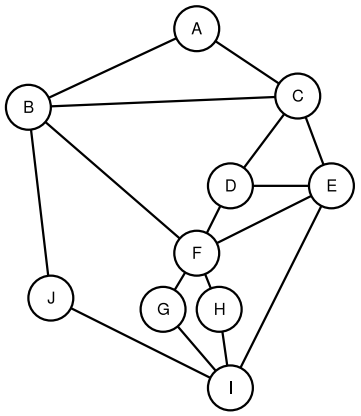
CSIS0270 Artificial Intelligence, 2003–2004
Assignment 2
Deadline Mar 20, 2004, 5:00pm.

This assignment contains written parts and programming parts. For the written parts, hand-in your answers to the assignment box (B3). For the programming parts, hand-in the source code of the programs that you write via the hand-in system of the department. You may also type your answer and send it to us using the handin system even for the written parts, but don't use a word document. Instead, just send us a plain text file that contains your answer (and represent graphical contents in a textual way).

1. ***CSP (35%)*** In the web page you can find a program that solves the *n*-queen problem. The program uses forward checking, without any other heuristic. It then reports the found solution, and also the number of times the forward checking routine removes a value from a variable.

   Add MRV and LCV to the program, with random tie breaking for both heuristics. Try the program several times with different seeds in order to find a solution for the *n*-queen problem for each $n \leq 70$. Report how much improvement you get.

2. ***Cycle Cutset and Tree Decomposition (15%)*** Consider the following map, to be colored using one of 7 colors, numbered 0 to 6. Some countries dislike certain colors, so not all colors can be used for each country. Furthermore, the colors are quite close to each other. We want to color our map so that neighbours have colors that differ by at least 2 (i.e., if the neighbour of a country is colored 1, the country cannot be colored 0, 1 or 2).



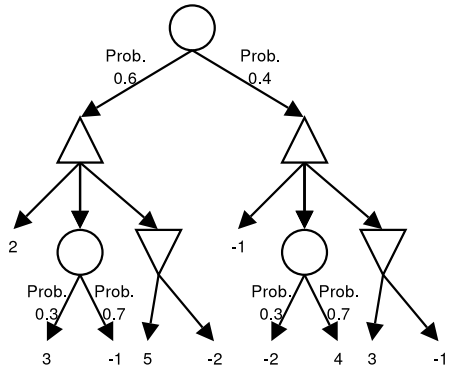
Acceptable colors
A: 2, 3, 5
B: 3, 4, 5, 6
C: 1, 4
D: 2, 3, 5, 6
E: 0, 1, 3, 5
F: 0, 4
G: 1, 2, 3
H: 3, 4, 6
I: 2, 3, 4, 5
J: 4, 5

   a. Find the minimum cycle cutset. Hence give a coloring of the graph, using the tree-CSP algorithm. Show all your steps.

   b. Find a tree decomposition which has a tree-width of 3 (i.e., at most 4 nodes in each component).

3. ***Alpha-Beta (35%)*** In the web page you can find a program that plays 4-in-a-row. Add a computer-play engine to it, which should use Alpha-Beta. Design a evaluation function so that you can cut-off at some depth rather than spending endless clock cycles to evaluate to the bottom of the game tree. Use at least one techinque in the lessons to increase the depth of Alpha-Beta that can be searched. Modify Alpha-Beta to add an element of chance, so that moves with the same evaluation are chosen at the same probability, moves that are slightly suboptimal are chosen with slightly less probability, and moves that are much worse are never chosen.

4. ***Incomplete information (15%)*** Games involving incomplete information can be difficult to play, even when the game tree is simple. In one such game, the player has to choose among three different actions, a, b and c. He has incomplete information of the current state: he estimates that the current state is either at A or B, but he is not sure. He pictures the decision tree as follows.



a. What is his decision if averaging over clairvoyance is used to evaluate the states?

b. Suppose his assessment of the game tree is accurate, and his opponent plays in an optimal way. Why it is possible that the decision of averaging over clairvoyance is not optimal? Illustrate your answer with an example in which the optimal choice is not the one as chosen by averaging over clairvoyance.