Computer Science 130B
Spring 2107
Homework #4

Due: DO NOT TURN IN, for your exercise only

CAVEAT: The first two problems should be solved by programming. It will be too tedious to solve them using a pencil and paper.

Problem 1 Given an $n \times n$ chessboard, a knight is placed on an arbitrary square with coordinate $(x, y), 1 \leq x, y \leq n$. The problem is to determine $n^2 - 1$ knight moves such that every square is visited once if such a sequence of moves exists. Write an algorithm to solve this problem. Note that if the problem is solvable, any one solution will suffice. There is no need to find ALL the solutions. Solve this problem for the particular instance of $n = 6$. Output the result, if solvable, as n^2 (x,y) coordinates of the knight's moves on the board, where $1 \leq x, y \leq 6$.

Problem 2 If you are given n households that want to adopt a dog and n dogs in a local shelter available for adoption and two $n \times n$ arrays P and Q such that $P(i, j)$ is the preference of household i for dog j and $Q(i, j)$ is the preference of dog i for household j. The preference is a coded as a zero or a positive number; the larger the number, the higher the preference. If household i is matched with dog j then $P(i, j) \times Q(j, i)$ is the "profit" gained by the particular match. A *pairing* is a set of matches in which every household is matched with a unique dog. Write an algorithm that finds a pairing such that the sum of the product of preferences (or profit) is maximized. Use your algorithm to solve the following instance of the problem. Again, pencil-and-paper solution can be really tedious and computer implementation is needed. Turn in both codes and results. Output the results as a sequence of n 2-tuple $(ID_{household}, ID_{dog}), 1 \leq ID_{household}, ID_{dog} \leq n$ and the profit gained by the proposed pairing.

$$P = \begin{bmatrix} 1 & 3 & 7 & 0 & 2 & 4 \\ 9 & 5 & 5 & 10 & 1 & 0 \\ 1 & 10 & 8 & 7 & 7 & 3 \\ 9 & 10 & 1 & 0 & 4 & 0 \\ 6 & 2 & 4 & 8 & 7 & 1 \\ 1 & 10 & 9 & 9 & 2 & 8 \end{bmatrix} \tag{1}$$

$$Q = \begin{bmatrix} 7 & 8 & 7 & 1 & 8 & 5 \\ 3 & 8 & 8 & 5 & 3 & 1 \\ 10 & 2 & 3 & 10 & 5 & 1 \\ 0 & 5 & 7 & 3 & 7 & 3 \\ 4 & 4 & 7 & 6 & 9 & 8 \\ 4 & 6 & 2 & 2 & 0 & 3 \end{bmatrix} \tag{2}$$

Problem 3 Let $W = (5, 10, 12, 15, 18)$ and $M = 35$. Find all possible subsets of W which sum to M. Draw the portion of the state space tree which is generated.