SAN FRANCISCO STATE UNIVERSITY
Computer Science Department

CSC510 Section 04 – Analysis of Algorithms
Algorithm Challenge 4: Backtracking

Instructor: Jose Ortiz

Full Name: ______________________________________
Student ID: ______________________________________

Assignment Instructions. Must read!

Note: Failure to follow the following instructions in detail will impact your grade negatively.

1. This algorithm challenge is worth 10%, and will be graded using a grading point scale where the maximum possible grade is 100 points. For instance, if your grade in this assignment is 85/100, then this is equivalent to 0.85*10%=8.5% of 10%

2. Take into account that in this type of assignments, I am more interested in the way you approach the problem rather than your final solution.

3. In this algorithm challenge, you don't need to write working code; only pseudocode

Problem Statement

Given a $N * N$ *Matrix* M filled with positive integers, find all the possible cells $M(i, j)$ where indexes i and j are unique and the sum of those cells is maximized or minimized for all the possible solutions found.

The formal definition of the problem is the following:

Let $\{P_1, P_2, ..., P_k, ..., P_n\}$ be a set of solutions for this problem where $P_k = \{M(i,j)_1 + M(i,j)_2 + M(i,j)_3 + ... + M(i,j)_{m-1} + M(i,j)_m\} = S$ is a set of coordinates for integers values in a matrix, and S the sum of those integers for that solution P_k. The S sum is valid only if:

1. All the indexes i *and* j for that sum of P_k are unique
2. The integer in $M(i, j)$ is not zero
3. Index j in $M(i, j)_x$ must be the same as index i in $M(i, j)_{x+1}$
4. Index i in $M(i, j)_1$ and index j in $M(i, j)_m$ must be zero for all the solutions P_k
5. A possible solution P_k is only considered an optimal solution if the sum S of all its integers is the minimum or the maximum sum S from all the solutions P_k

For example, given the following matrix M filled with integers and zeros find all the possible results that met the above conditions.

0	3	6	0
3	0	2	5
6	2	0	1
0	5	1	0

All possible solutions are:

1. $P = \{M[0][1] + M[1][3] + M[3][2] + M[2][0]\} = 15$
2. $P = \{M[0][2] + M[2][3] + M[3][1] + M[1][0]\} = 15$

As you can see, solutions 1 and 2 met all the conditions above.

Your work starts here

1. (40 points) Create a backtracking algorithm (using a state space tree) to find all the solutions from the example above. Note that the below matrix M is the same as the one from the example

0	3	6	0
3	0	2	5
6	2	0	1
0	5	1	0

2. (20 points) Use the backtracking algorithm created in part 1 to find all the solutions for the following matrix M:

0	7	1	0	0	8
7	0	5	0	9	6
1	5	0	4	0	0
0	0	4	0	2	0
0	9	0	2	0	3
8	6	0	0	3	0

3. (20 points) Based on your backtracking algorithm, create pseudecode that finds all the solutions for any given matrix M. Note that your pseudocode can be either iterative or recursive.

4. (20 points) Find the time and space complexity of your backtracking algorithm. Note that if your pseudocode is recursive, you may need a recursive relation first, and then find its time complexity using the master theorem. On the other hand, if your pseudecode is iterative, then you can find its time complexity using a step counting approach.