CSCI 511 Analysis of Algorithms II **Winter 2013 - Johnson**

David Palzer
Midterm 2

1. The number of candidate decompositions is 2^n where n is the number of vertices in the polygon. This is because for each combination you would like to try there are two subproblems that you would encounter with a total of n-1 vertices between each. There are n combinations of these subproblems. This results in recursing and finding that there are a total of 2^n different leaves that you could end up at at the bottom of the recusion tree and an 2^n sized problem.

2. The dynamic programming algorithm that I have discovered takes the list of vertices and and array of size NxN. Each vertices has n subproblems i.e. t(1,k) + t(k+1,j), where k loops from i to j-1. The algorithm splits each problem into the n subproblems it has and then solves them one by one. Once a subproblem has been solved it is then recorded into the NxN array. Before recursing on a problem, the algorithm first checks to see if it has been solved before. If it has then it goes into the array and grabs the solution rather than solving it again and wasting time. This reduces the number of leaves on the tree from 2^n to n^3. The work of combining these results to find the solution is contained within the recursion so does not add to the run time. The same holds true for recording the different chords and piling them together. This results in an n^3 run time solution for finding the chords and minimal decomposition of a convex polygon into triangles.

3. See attached for code and output.