22C:31 Algorithms
Second Exam (100 points)
Closed books and notes (except two sheets of notes)

When asking for designing an algorithm, both English description and pseudo-code are required. When asking for an efficient algorithm, the score will depend on how efficient your algorithm is.

1. (20 points) Please design an efficient algorithm which decides if a given undirected graph $G = (V, E)$ has a triangle (cycles of length 3).

2. (30 points) Given a string $S$ of length $n$, design an efficient algorithm which computes the longest palindromic subsequence of $S$ (not necessarily consecutive). A palindromic string is one which reads the same front and back. For example, the string RAECEDCAUR has inside it palindromes RR, RAECAR, RACECAR, etc. At first, you provide a recursive definition of the length of a longest palindromic subsequence of a string. Based on this recursive definition, you provide a dynamic programming algorithm. You receive the full score if your algorithm is correct and runs in time $O(n^2)$.

3. (50 points) Hawkeye golf team has $2n$ players and each player has a numeric talent score, which is a non-negative number. Coach wants to split the team into two groups of $n$ players each such that the total talent score of one group is as close as possible to the total talent score of the other group. The above description is the optimization version of the golf team problem. The decision version of this problem is as follows: Given a set $S$ of $2n$ non-negative numbers and an integer $K$. Can we partition $S$ into $S_1$ and $S_2$ such that $|S_1| = |S_2| = n$ and $|\text{sum}(S_1) - \text{sum}(S_2)| \leq K$? Here $\text{sum}(X)$ denotes the total sum of numbers in the set $X$.
   a. (25 points) Show formally that the decision version of the golf team problem is NP-complete.
   b. (25 points) If we have a polynomial time algorithm $D(S, K)$ for the decision version of the golf team problem, show that the optimization version of the problem can be solved in polynomial time using $D(S, K)$. 