CS 341 EXAM

Total: 100 marks. **Aids Allowed:** One letter sized piece of paper with notes.
If you answer questions out of order, please indicate that clearly so we don't miss any.

1. (20 marks) Recall that 2 sorted arrays with n and m elements respectively can be merged into a single sorted array of $n+m$ elements in $O(n+m)$ time. Suppose you have k sorted arrays each with n elements, and you want to combine them into a single sorted array of kn elements.

   (a) (10 marks) One approach is to merge the first two arrays, then merge the result with the third array, then merge the result of that with the fourth, etc. Analyze the run time of this method in terms of n and k.

   (b) (10 marks: 5 for the algorithm, 5 for analysis) Give a more efficient method using divide and conquer.

2. Prove that the following problem, called "Cheapest 3-SAT" is NP complete: Given a Boolean formula F in 3-SAT form with n and m clauses, and a number $k \in \mathbb{N}$, is there a satisfying truth-value assignment with at most k of the n variables set to True?

   The "known" NP-complete problems you may use for your reduction are: 3-SAT, Independent Set, Vertex Cover, Set Cover, Hamiltonian Cycle, Hamiltonian Path, TSP, and Subset Sum.

3. (20 marks) This question is about finding a shortest cycle in a graph.

   (a) (10 marks) Consider the case of an undirected graph with edge weights all equal to 1. Here is an attempt at an algorithm. Run Depth First Search on the graph. Every time a back edge, say (u,v) is discovered, it forms a cycle together with the path in the DFS tree from v to u. If this cycle C is shortest one found so far, then save C as the shortest cycle found so far.

   Show that this algorithm does not always find a shortest cycle–give a counterexample and a brief explanation of why the method fails.

   (b) (10 marks) Give an efficient algorithm for the case of directed graph $G=(V,E)$ with positive edge weights $w : E \to \mathbb{N}$. If the graph has no cycle, your algorithm should say so. Aim for a run time of $O(n^3)$ where $n = |V|$. Justify correctness, and analyze run time. (You may use algorithms for class.)

4. (20 marks) Consider the following BALANCE problem: Given numbers $a_1, a_2, ..., a_n$ with $a_i \in \mathbb{N}$, can you split $1, 2, ..., n$ into two sets S_1 and S_2 with $\sum_{i \in S_1} a_i = \sum_{i \in S_2} a_i$? – i.e. can you put the numbers into two piles whose sums balance?

   (a) (2 marks) Show that the answer is Yes for the input 2,1,5,3,3. Show that the answer is No for the input 5,3,3,6.

   (b) (10 marks) Prove that the Balance problem is NP-complete.

   The "known" NP-complete problems you may use for your reduction are: 3-SAT, Independent Set, Vertex Cover, Set Cover, Hamiltonian Cycle, Hamiltonian Path, TSP, and Subset Sum.

   (c) (8 marks) Give a backtracking algorithm for the Balance problem and analyze its run time. You may describe your algorithm at a high level, and in fact, describing the general structure of the search tree is already worth most of the marks.

5. (10 marks) Show that the following problem is undecidable: Given a program P that reads no input, does it output "Hello World"?

   The "known" undecidable problems you may use for your reduction are: the Halting Problem and the Halt-no-Input problem.

6. (10 marks) True or False (or Open). In all cases, justify briefly.

   (a) $n^{1+\epsilon}$ is in $O(n \log n)$

   (b) There is a backtracking algorithm for the Halting Problem.

   (c) The following problem is NP-complete: Given a graph, does it have an independent set of size 3?

   (d) The Travelling Salesman Problem cannot be solved in polynomial time.

   (e) The following problem is undecidable. Given a program P and input w, and number $k \in \mathbb{N}$, does P halt on input w within k steps?

7. (10 marks; this question may be a bit more challenging) A palindrome is a string that reads the same from left to right as it does from right to left. For example, "OO" and "MADAM" are palindromes. Give an $O(n^2)$ time dynamic programming algorithm that takes as input a string of length n and returns the maximum length of a substring that is a palindrome.

   For example, for the "AABACCAAA" the output is 4 because "ACCA" is the longest substring that is a palindrome.