CS 473g: Algorithms, Fall 2007
Homework 1, Practice Problems

Practice Problems

1. NP-Completeness collection
   [20 Points]

   Prove that the following problems are NP-Complete.

   A. **[4 Points]**

   Problem: MINIMUM SET COVER

Instance: Collection C of subsets of a finite set S and an integer k. *Question:* Are there k sets $S_1, \ldots, S_k$ in C such that $S \subseteq \cup_{i=1}^{k} S_i$?

   B. **[4 Points]**

   Problem: BIN PACKING

Instance: Finite set U of items, a size $s(u) \in \mathbb{Z}^+$ for each $u \in U$, an integer bin capacity B, and a positive integer K. *Question:* Is there a partition of U into disjoint sets $U_1, \ldots, U_K$ such that the sum of the sizes of the items inside each U_i is B or less?

   C. **[4 Points]**

   Problem: TILING

Instance: Finite set $\mathcal{R}$ of rectangles and a rectangle R in the plane. *Question:* Is there a way of placing the rectangles of $\mathcal{R}$ inside R, so that no pair of the rectangles intersect, and all the rectangles have their edges parallel of the edges of R?

   D. **[4 Points]**

   Problem: HITTING SET

Instance: A collection C of subsets of a set S, a positive integer K. *Question:* Does S contain a *hitting set* for C of size K or less, that is, a subset $S' \subseteq S$ with $\lvert S' \rvert \leq K$ and such that S' contains at least one element from each subset in C.

   E. **[4 Points]**

   Problem: Max Degree Spanning Tree

Instance: Graph $G = (V, E)$ and integer k *Question:* Does G contains a spanning tree T where every node in T is of degree at most k?

2. Partition **[10 Points]**

   The Partition satyr, the uncle of the deduction fairy, had visited you on winter break and gave you, as a token of appreciation, a black-box that can solve Partition in polynomial time (note that this black box solves the decision problem). Let S be a given set of n integer numbers.

Describe a polynomial time algorithm that computes, using the black box, a partition of S if such a solution exists. Namely, your algorithm should output a subset $T \subseteq S$, such that

$$\sum_{s \in T} s = \sum_{s \in S \setminus T} s.$$

3. From decision to explicit solution.
[20 Points]

   A. **[10 Points]** The Partition satyr, the uncle of the deduction fairy, had visited you on winter break and gave you, as a token of appreciation, a black-box that can solve Partition in polynomial time (note that this black box solves the decision problem). Let S be a given set of n integer numbers. Describe a polynomial time algorithm that computes, using the black box, a partition of S if such a solution exists. Namely, your algorithm should output a subset $T \subseteq S$, such that

$$\sum_{s \in T} s = \sum_{s \in S \setminus T} s.$$

   B. **[10 Points]** The deduction fairy turned out to have a large family, and her niece came and visited you and left you with a black box that can solve the MINIMUM SET COVER in polynomial time. Show a polynomial time algorithm for solving the minimum set cover problem (i.e., it outputs the sets in the smallest cover of the given set) using this black box.

   (You should verify, for fun, what problems in the previous exercise can be solved in polynomial time given a black box for the decision problem.)

4. Partition revisited **[10 Points]**

   Let S be an instance of partition, such that $n = |S|$, and $M = \max_{s \in S} s$. Show a polynomial time (in n and M) algorithm that solves partition.

5. Why Mike can not get it. **[10 Points]**

Problem: Not-3SAT

Instance: A 3CNF formula F *Question:* Is F not satisfiable? (Namely, for all inputs for F, it evaluates to `FALSE`.)

   (a) Prove that Not-3SAT is $co - NP$.

   (b) Here is a proof that Not-3SAT is in NP: If the answer to the given instance is `Yes`, we provide the following proof to the verifier: We list every possible assignment, and for each assignment, we list the output (which is FALSE). Given this proof, of length L, the verifier can easily verify it in polynomial time in L. QED.

   What is wrong with this proof?

   (c) Show that given a black-box that can solves Not-3SAT, one can find the satisfying assignment of a formula F in polynomial time, using polynomial number of calls to the black-box (if such an assignment exists).

6. NP-Completeness Collection **[20 Points]**

Prove that the following problems are NP-Complete.

Problem: MINIMUM SET COVER (a)

> *Instance:* Collection C of subsets of a finite set S and an integer k.
> *Question:* Are there k sets $S_1, \ldots, S_k$ in C such that $S \subseteq \cup_{i=1}^{k} S_i$?

Problem: HITTING SET (b)

> *Instance:* A collection C of subsets of a set S, a positive integer K.
> *Question:* Does S contain a *hitting set* for C of size K or less, that is, a subset $S' \subseteq S$ with $|S'| \leq K$ and such that S' contains at least one element from each subset in C.

Problem: Hamiltonian Path (c)

> *Instance:* Graph $G = (V, E)$
> *Question:* Does G contains a Hamiltonian path? (Namely a path that visits all vertices of G.)

Problem: Max Degree Spanning Tree (d)

> *Instance:* Graph $G = (V, E)$ and integer k
> *Question:* Does G contains a spanning tree T where every node in T is of degree at most k?

7. Independence **[10 Points]**

Let $G = (V, E)$ be an undirected graph over n vertices. Assume that you are given a numbering $\pi : V \to \{1, \ldots, n\}$ (i.e., every vertex have a unique number), such that for any edge $ij \in E$, we have $|\pi(i) - \pi(j)| \leq 20$.

Either prove that it is NP-Hard to find the largest independent set in G, or provide a polynomial time algorithm.

8. Partition **[10 Points]**

We already know the following problem is NP-Complete

Problem: SUBSET SUM

> *Instance:* A finite set A and a "size" $s(a) \in \mathbb{Z}^+$ for each $a \in A$, an integer B.
> *Question:* Is there a subset $A' \subseteq A$ such that $\sum_{a \in A'} s(a) = B$?

Now let's consider the following problem:

Problem: PARTITION

> *Instance:* A finite set A and a "size" $s(a) \in \mathbb{Z}^+$ for each $a \in A$.
> *Question:* Is there a subset $A' \subseteq A$ such that
>
> $$\sum_{a \in A'} s(a) = \sum_{a \in A \setminus A'} s(a)?$$

Show that PARTITION is NP-Complete.

9. MINIMUM SET COVER **[15 Points]**

Problem: MINIMUM SET COVER

Instance: Collection C of subsets of a finite set S and an integer k. *Question:* Are there k sets $S_1, \ldots, S_k$ in C such that $S \subseteq \cup_{i=1}^{k} S_i$?

(a) **[5 Points]** Prove that MINIMUM SET COVER problem is NP-Complete

(b) **[5 Points]** Prove that the following problem is NP-Complete.

Problem: HITTING SET

Instance: A collection C of subsets of a set S, a positive integer K. *Question:* Does S contain a *hitting set* for C of size K or less, that is, a subset $S' \subseteq S$ with $|S'| \leq K$ and such that S' contains at least one element from each subset in C.

(c) **[5 Points]** *Hitting set on the line*
Given a set $\mathcal{I}$ of n intervals on the real line, show a $O(n \log n)$ time algorithm that computes the smallest set of points X on the real line, such that for every interval $I \in mathcal I$ there is a point $p \in X$, such that $p \in I$.

10. BIN PACKING **[10 Points]**

Problem: BIN PACKING

Instance: Finite set U of items, a size $s(u) \in \mathbb{Z}^+$ for each $u \in U$, an integer bin capacity B, and a positive integer K. *Question:* Is there a partition of U into disjoint sets $U_1, \ldots, U_K$ such that the sum of the sizes of the items inside each U_i is B or less?

(a) **[5 Points]** Show that the BIN PACKING problem is NP-Complete

(b) **[5 Points]** Show that the following problem is NP-Complete.

Problem: TILING

Instance: Finite set $\mathcal{RECTS}$ of rectangles and a rectangle R in the plane. *Question:* Is there a way of placing all the rectangles of $\mathcal{RECTS}$ inside R, so that no pair of the rectangles intersect in their interior, and all the rectangles have their edges parallel of the edges of R?

11. KNAPSACK **[15 Points]**

(a) **[5 Points]** Show that the following problem is NP-Complete.

Problem: KNAPSACK

Instance: A finite set U, a "size" $s(u) \in \mathbb{Z}^+$ and a "value" $v(u) \in \mathbb{Z}^+$ for each $u \in U$, a size constraint $B \in \mathbb{Z}+$, and a value goal $K \in \mathbb{Z}+$. *Question:* Is there a subset $U' \subseteq U$ such that $\sum_{u \in U'} s(u) \leq B$ and $\sum_{u \in U'} v(u) \geq B$.

(b) **[5 Points]** Show that the following problem is NP-Complete.

Problem: MULTIPROCESSOR SCHEDULING

> *Instance:* A finite set A of "tasks", a "length" $l(a) \in \mathbb{Z}^+$ for each $a \in A$, a number $m \in \mathbb{Z}^+$ of "processors", and a "deadline" $D \in \mathbb{Z}^+$.
> *Question:* Is there a partition $A = A_1 \bigcup A_2 \bigcup \cdots \bigcup A_m$ of A into m disjoint sets such that $max\{\sum_{a \in A_i} l(a) : 1 \leq i \leq m\} \leq D$?

(c) Scheduling with profits and deadlines **[5 Points]**

Suppose you have one machine and a set of n tasks $a_1, a_2, ..., a_n$. Each task a_j has a processing time t_j, a profit p_j, and a deadline d_j. The machine can process only one task at a time, and task a_j must run uninterruptedly for t_j consecutive time units to complete. If you complete task a_j by its deadline d_j, you receive a profit p_j. But you receive no profit if you complete it after its deadline. As an optimization problem, you are given the processing times, profits and deadlines for a set of n tasks, and you wish to find a schedule that completes all the tasks and returns the greatest amount of profit.

i. **[3 Points]** State this problem as a decision problem.

ii. **[2 Points]** Show that the decision problem is NP-complete.

12. Vertex Cover

Problem: VERTEX COVER

> *Instance:* A graph $G = (V, E)$ and a positive integer $K \leq |V|$.
> *Question:* Is there a *vertex cover* of size K or less for G, that is, a subset $V' \subseteq V$ such that $|V'| \leq K$ and for each edge $\{u, v\} \in E$, at least one of u and v belongs to V'?

(a) Show that VERTEX COVER is NP-Complete. Hint: Do a reduction from INDEPENDENT SET to VERTEX COVER.

(b) Show a polynomial approximation algorithm to the Vertex-Cover problem which is a factor 2 approximation of the optimal solution. Namely, your algorithm should output a set $X \subseteq V$, such that X is a vertex cover, and $|C| \leq 2K_{opt}$, where K_{opt} is the cardinality of the smallest vertex cover of G.[1]

(c) Present a linear time algorithm that solves this problem for the case that G is a tree.

(d) For a constant k, a graph G is k-separable, if there are k vertices of G, such that if we remove them from G, each one of the remaining connected components has at most $(2/3)n$ vertices, and furthermore each one of those connected components is also k-separable. (More formally, a graph $G = (V, E)$ is k-separable, if for any subset of vertices $S \subseteq V$, there exists a subset $M \subseteq S$, such that each connected component of $G_{S \setminus M}$ has at most $(2/3)|S|$ vertices, and $|M| \leq k$.)

Show that given a graph G which is k-separable, one can compute the optimal VERTEX COVER in $n^{O(k)}$ time.

13. Bin Packing

[1]It was very recently shown (I. Dinur and S. Safra. On the importance of being biased. Manuscript. `http://www.math.ias.edu/~iritd/mypapers/vc.pdf`, 2001.) that doing better than 1.3600 approximation to VERTEX COVER is NP-Hard. In your free time you can try and improve this constant. Good luck.

Problem: BIN PACKING

> *Instance:* Finite set U of items, a size $s(u) \in \mathbb{Z}^+$ for each $u \in U$, an integer bin capacity B, and a positive integer K.
> *Question:* Is there a partition of U int disjoint sets $U_1, \ldots, U_K$ such that the sum of the sizes of the items inside each U_i is B or less?

(a) Show that the BIN PACKING problem is NP-Complete

(b) In the optimization variant of BIN PACKING one has to find the minimum number of bins needed to contain all elements of U. Present an algorithm that is a factor two approximation to optimal solution. Namely, it outputs a partition of U into M bins, such that the total size of each bin is at most B, and $M \leq k_{opt}$, where k_{opt} is the minimum number of bins of size B needed to store all the given elements of U.

(c) Assume that B is bounded by an integer constant m. Describe a polynomial algorithm that computes the solution that uses the minimum number of bins to store all the elements.

(d) Show that the following problem is NP-Complete.

Problem: TILING

> *Instance:* Finite set $\mathcal{R}$ of rectangles and a rectangle R in the plane.
> *Question:* Is there a way of placing the rectangles of $\mathcal{R}$ inside R, so that no pair of the rectangles intersect, and all the rectangles have their edges parallel of the edges of R?

(e) Assume that $\mathcal{R}$ is a set of squares that can be arranged as to tile R completely. Present a polynomial time algorithm that computes a subset $\mathcal{T} \subseteq \mathcal{R}$, and a tiling of $\mathcal{T}$, so that this tiling of $\mathcal{T}$ covers, say, 10% of the area of R.

14. Minimum Set Cover

Problem: MINIMUM SET COVER

> *Instance:* Collection C of subsets of a finite set S and an integer k.
> *Question:* Are there k sets $S_1, \ldots, S_k$ in C such that $S \subseteq \cup_{i=1}^{k} S_i$?

(a) Prove that MINIMUM SET COVER problem is NP-Complete

(b) The greedy approximation algorithm for MINIMUM SET COVER, works by taking the largest set in $X \in C$, remove all all the elements of X from S and also from each subset of C. The algorithm repeat this until all the elements of S are removed. Prove that the number of elements not covered after k_{opt} iterations is at most $n/2$, where k_{opt} is the smallest number of sets of C needed to cover S, and $n = |S|$.

(c) Prove the greedy algorithm is $O(\log n)$ factor optimal approximation.

(d) Prove that the following problem is NP-Complete.

Problem: HITTING SET

> *Instance:* A collection C of subsets of a set S, a positive integer K.
> *Question:* Does S contain a *hitting set* for C of size K or less, that is, a subset $S' \subseteq S$ with $|S'| \leq K$ and such that S' contains at least one element from each subset in C.

(e) Given a set $\mathcal{I}$ of n intervals on the real line, show a $O(n \log n)$ time algorithm that computes the smallest set of points X on the real line, such that for every interval $I \in \mathcal{I}$ there is a point $p \in X$, such that $p \in I$.

15. k-Center Problem

Problem: k-CENTER

> *Instance:* A set P of n points in the plane, and an integer k and a radius r.
> *Question:* Is there a cover of the points of P by k disks of radius (at most) r?

(a) Describe an $n^{O(k)}$ time algorithm that solves this problem.

(b) There is a very simple and natural algorithm that achieves a 2-approximation for this cover: First it select an arbitrary point as a center (this point is going to be the center of one of the k covering disks). Then it computes the point that it furthest away from the current set of centers as the next center, and it continue in this fashion till it has k-points, which are the resulting centers. The smallest k equal radius disks centered at those points are the required k disks.

Show an implementation of this approximation algorithm in $O(nk)$ time.

(c) Prove that that the above algorithm is a factor two approximation to the optimal cover. Namely, the radius of the disks output $\leq 2r_{opt}$, where r_{opt} is the smallest radius, so that we can find k-disks that cover the point-set.

(d) Provide an ε-approximation algorithm for this problem. Namely, given k and a set of points P in the plane, your algorithm would output k-disks that cover the points and their radius is $\leq (1+\varepsilon)r_{opt}$, where r_{opt} is the minimum radius of such a cover of P.

(e) Prove that dual problem r-DISK-COVER problem is NP-Hard. In this problem, given P and a radius r, one should find the smallest number of disks of radius r that cover P.

(f) Describe an approximation algorithm to the r-DISK COVER problem. Namely, given a point-set P and a radius r, outputs k disks, so that the k disks cover P and are of radius r, and $k = O(k_{opt})$, where k_{opt} is the minimal number of disks needed to cover P by disks of radius r.

16. MAX 3SAT Problem

Problem: MAX SAT

> *Instance:* Set U of variables, a collection C of disjunctive clauses of literals where a literal is a variable or a negated variable in U.
> *Question:* Find an assignment that maximized the number of clauses of C that are being satisfied.

(a) Prove that MAX SAT is NP-Hard.

(b) Prove that if each clause has exactly three literals, and we randomly assign to the variables values 0 or 1, then the expected number of satisfied clauses is $(7/8)M$, where $M = |C|$.

(c) Show that for any instance of MAX SAT, where each clause has exactly three different literals, there exists an assignment that satisfies at least 7/8 of the clauses.

(d) Let (U, C) be an instance of MAX SAT such that each clause has $\geq 10 \cdot \log n$ distinct variables, where n is the number of clauses. Prove that there exists a satisfying assignment. Namely, there exists an assignment that satisfy all the clauses of C.

17. Complexity

    (a) Prove that P $\subseteq$ co-NP.

    (b) Show that if NP $\neq$ co-NP, then *every* NP-complete problem is *not* a member of co-NP.

18. 2-CNF-SAT
    Prove that deciding satisfiability when all clauses have at most 2 literals is in P.

19. Graph Problems

    (a) LONGEST-PATH
    Show that the problem of deciding whether an unweighted undirected graph has a path of length greater than k is NP-complete.

20. PARTITION, SUBSET-SUM
    PARTITION is the problem of deciding, given a set of numbers, whether there exists a subset whose sum equals the sum of the complement, i.e. given $S = s_1, s_2 \ldots, s_n$, does there exist a subset S' such that $\sum_{s \in S'} s = \sum_{t \in S - S'} t$. SUBSET-SUM is the problem of deciding, given a set of numbers and a target sum, whether there exists a subset whose sum equals the target, i.e. given $S = s_1, s_2 \ldots, s_n$ and k, does there exist a subset S' such that $\sum_{s \in S'} s = k$. Give two reduction, one in both directions.

21. 3SUM
    Describe an algorithm that solves the following problem as quickly as possible: Given a set of n numbers, does it contain three elements whose sum is zero? For example, your algorithm should answer TRUE for the set $\{-5, -17, 7, -4, 3, -2, 4\}$, since $-5+7+(-2) = 0$, and FALSE for the set $\{-6, 7, -4, -13, -2, 5, 13\}$.

22. Consider finding the median of 5 numbers by using only comparisons. What is the exact worst case number of comparisons needed to find the median. Justify (exhibit a set that cannot be done in one less comparisons). Do the same for 6 numbers.

23. EXACT-COVER-BY-4-SETS
    The EXACT-COVER-BY-3-SETS problem is defines as the following: given a finite set X with $|X| = 3q$ and a collection C of 3-element subsets of X, does C contain an *exact cover* for X, that is, a subcollection $C' \subseteq C$ such that every element of X occurs in exactly one member of C'?

    Given that EXACT-COVEqR-BY-3-SETS is NP-complete, show that EXACT-COVER-BY-4-SETS is also NP-complete.

24. PLANAR-3-COLOR
    Using 3-COLOR, and the 'gadget' in figure 24, prove that the problem of deciding whether a planar graph can be 3-colored is NP-complete. Hint: show that the gadget can be 3-colored, and then replace any crossings in a planar embedding with the gadget appropriately.

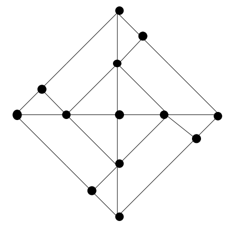

Figure 1: Gadget for PLANAR-3-COLOR.

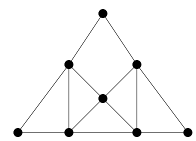

Figure 2: Gadget for DEGREE-4-PLANAR-3-COLOR.

25. DEGREE-4-PLANAR-3-COLOR
Using the previous result, and the 'gadget' in figure 25, prove that the problem of deciding whether a planar graph with no vertex of degree greater than four can be 3-colored is NP-complete. Hint: show that you can replace any vertex with degree greater than 4 with a collection of gadgets connected in such a way that no degree is greater than four.

26. Poly time subroutines can lead to exponential algorithms
Show that an algorithm that makes at most a constant number of calls to polynomial-time subroutines runs in polynomial time, but that a polynomial number of calls to polynomial-time subroutines may result in an exponential-time algorithm.

27. (a) Prove that if G is an undirected bipartite graph with an odd number of vertices, then G is nonhamiltonian. Give a polynomial time algorithm algorithm for finding a **hamiltonian cycle** in an undirected bipartite graph or establishing that it does not exist.
    (b) Show that the **hamiltonian-path** problem can be solved in polynomial time on directed acyclic graphs by giving an efficient algorithm for the problem.
    (c) Explain why the results in previous questions do not contradict the facts that both HAM-CYCLE and HAM-PATH are NP-complete problems.

28. Consider the following pairs of problems:
    (a) MIN SPANNING TREE and MAX SPANNING TREE
    (b) SHORTEST PATH and LONGEST PATH
    (c) TRAVELING SALESMAN PROBLEM and VACATION TOUR PROBLEM (the longest tour is sought).
    (d) MIN CUT and MAX CUT (between s and t)
    (e) EDGE COVER and VERTEX COVER
    (f) TRANSITIVE REDUCTION and MIN EQUIVALENT DIGRAPH

(all of these seem dual or opposites, except the last, which are just two versions of minimal representation of a graph).

Which of these pairs are polytime equivalent and which are not? Why?

29. GRAPH-ISOMORPHISM
Consider the problem of deciding whether one graph is isomorphic to another.
    (a) Give a brute force algorithm to decide this.
    (b) Give a dynamic programming algorithm to decide this.
    (c) Give an efficient probabilistic algorithm to decide this.
    (d) Either prove that this problem is NP-complete, give a poly time algorithm for it, or prove that neither case occurs.

30. Prove that PRIMALITY (Given n, is n prime?) is in NP $\cap$ co-NP. Hint: co-NP is easy (what's a certificate for showing that a number is composite?). For NP, consider a certificate involving primitive roots and recursively their primitive roots. Show that knowing this tree of primitive roots can be checked to be correct and used to show that n is prime, and that this check takes poly time.

31. How much wood would a woodchuck chuck if a woodchuck could chuck wood?