Chapter 33

Exercises - NP Completeness

By Sariel Har-Peled, October 3, 2013[1] Version: 1.02

33.1 Equivalence of optimization and decision problems

33.1.1 Beware of Greeks bearing gifts

(The expression "beware of Greeks bearing gifts" is Based on Virgil's Aeneid: "Quidquid id est, timeo Danaos et dona ferentes", which means literally "Whatever it is, I fear Greeks even when they bring gifts.".)

The reduction faun, the brother of the Partition satyr, came to visit you on labor day, and left you with two black boxes.

(A) (10 PTS.) The first black box, was a black box that can solves the following decision problem in polynomial time:

Minimum Test Collection

Instance: A finite set A of "possible diagnoses," a collection C of subsets of A, representing binary "tests," and a positive integer $J \leq |C|$.
Question: Is there a subcollection $C' \subseteq C$ with $|C'| \leq J$ such that, for every pair a_i, a_j of possible diagnoses from A, there is some test $c \in C'$ for which $|\{a_i, a_j\} \cap c| = 1$ (that is, a test c that "distinguishes" between a_i and a_j)?

Show how to use this black box, how to solve in polynomial time the optimization version of this problem (i.e., finding and outputting the smallest possible set C').

(B) (10 PTS.)
The second box was a black box for solving Subgraph Isomorphism.

Subgraph Isomorphism

Instance: Two graphs, $G = (V_1, E_1)$ and $H = (V_2, E_2)$.
Question: Does G contain a subgraph *isomorphic* to H, that is, a subset $V \subseteq V_1$ and a subset $E \subseteq E_1$ such that $|V| = |V_2|$, $|e| = |E_2|$, and there exists a one-to-one function $f : V_2 \to V$ satisfying $\{u, v\} \in E_2$ if and only if $\{f(u), f(v)\} \in E$?

Show how to use this black box, to compute the subgraph isomorphism (i.e., you are given G and H, and you have to output f) in polynomial time.

33.1.2 Partition

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

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

33.2 Showing problems are NP-Complete

33.2.1 Graph Isomorphisms

(A) (5 PTS.) Show that the following problem is NP-COMPLETE.

SUBGRAPH ISOMORPHISM

reduction of clique to this, where we can create a trivial complete graph of k vertices and ask the subgraph isomorphism question on it

Instance: Graphs $G = (V_1, E_1), H = (V_2, E_2)$.
Question: Does G contain a subgraph isomorphic to H, i.e., a subset $V \subseteq V_1$ and a subset $E \subseteq E_1$ such that $|V| = |V_2|$, $|E| = |E_2|$, and there exists a one-to-one function $f : V_2 \to V$ satisfying $\{u, v\} \in E_2$ if and only if $\{f(u), f(v)\} \in E$?

(B) (5 PTS.) Show that the following problem is NP-COMPLETE.

LARGEST COMMON SUBGRAPH

again reduction from clique only. clique takes an input graph G and a no k. for H we will make a complete graph of k vertices ask the question with this graph and K = k*k

Instance: Graphs $G = (V_1, E_1), H = (V_2, E_2)$, positive integer K.
Question: Do there exists subsets $E_1' \subseteq E_1$ and $E_2' \subseteq E_2$ with $|E_1'| = |E_2'| \geq K$ such that the two subgraphs $G' = (V_1, E_1')$ and $H' = (V_2, E_2')$ are isomorphic?

33.2.2 NP Completeness collection

(A) (5 PTS.)

MINIMUM SET COVER

can reduce vertex cover to this
the finite set S is the set of all edges
and each subset represent a vertex, and the edges that it is able to cover
is there a vertex cover of size k => is there a set cover of size k?

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) (5 PTS.)

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?

(C) (5 PTS.)

reduction of partition to bin packing
size of element u is its value
K = 2
B = sum of the values of all elements / 2

TILING

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

(D) (5 PTS.)

HITTING SET

vertex cover can be reduced to this

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.

33.2.3 LONGEST-PATH

hamiltonian path problem can be reduced to this

Show that the problem of deciding whether an unweighted undirected graph has a path of length greater than k is NP-COMPLETE.

so i need to convert the given X, C into X' and C' such that the exact-cover-by-3-sets problem stands solved
consider the following : X' will contain random q more elements
now C' = C*q, as in the product of C with the set of q elements that got added

33.2.4 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-COVER-BY-3-SETS is NP-COMPLETE, show that EXACT-COVER-BY-4-SETS is also NP-COMPLETE.

33.3 Solving special subcases of NP-Complete problems in polynomial time

33.3.1 Subset Sum

so i need to convert the given X, C into X' and C' such that the exact-cover-by-3-sets problem stands solved
consider the following : X' will contain random q more elements
now C' = C*q, as in the product of C with the set of q elements that got added

Subset Sum

Instance: S - set of positive integers, t: - an integer number
Question: Is there a subset $X \subseteq S$ such that

$$\sum_{x \in X} x = t \ ?$$

Given an instance of Subset Sum, provide an algorithm that solves it in polynomial time in n, and M, where $M = \max_{s \in S} s$. Why this does not imply that $P = NP$?

33.3.2 2SAT

Given an instance of 2SAT (this is a problem similar to 3SAT where every clause has at most two variables), one can try to solve it by backtracking.

(A) (1 PTS.) Prove that if a formula F' is not satisfiable, and F is formed by adding clauses to F', then the formula F is not satisfiable. (Duh?)
We refer to F' as a *subformula* of F.

(B) (3 PTS.) Given an assignment $x_i \leftarrow b$ to one of the variables of a 2SAT instance F (where b is either 0 or 1), describe a polynomial time algorithm that computes a subformula F' of F, such that (i) F' does not have the variable x_i in it, (ii) F' is a 2SAT formula, (iii) F' is satisfiable iff there is a satisfying assignment for F with $x_i = b$, and (iv) F' is a subformula of F.
How fast is your algorithm?

(C) (6 PTS.) Describe a polynomial time algorithm that solves the 2SAT problem (using (b)). How fast is your algorithm?

33.3.3 2-CNF-SAT

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

33.3.4 Hamiltonian Cycle Revisited

Let C_n denote the cycle graph over n vertices (i.e., $V(C_n) = \{1, \ldots, n\}$, and $E(C_n) = \{\{1,2\}, \{2,3\}, \ldots, \{n-1, n$
Let C_n^k denote the graph where $\{i, j\} \in E(C_n^k)$ iff i and j are in distance at most k in C_n.

Let G be a graph, such that G is a subgraph of C_n^k, where k is a small constant. Describe a polynomial time algorithm (in n) that outputs a Hamiltonian cycle if such a cycle exists in G. How fast is your algorithm, as a function of n and k?

33.3.5 Partition revisited

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.

33.3.6 Why Mike can not get it.

(10 PTS.)

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).

33.3.7 NP-Completeness Collection

(20 PTS.) Prove that the following problems are NP-Complete.

reducing vertex cover to this
S is the set of all edges
each set in collection C corresponds to a vertex(the edges it will cover)

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 $S \subseteq \cup_{i=1}^{k} S_i$?

HITTING SET

vertex cover can be reduced to dominating set
dominating set can be reduced to this

(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.

Hamiltonian Path

3sat can be reduced to hamiltonian path

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

Max Degree Spanning Tree

hamiltonian path can be reduced to this? wherein k becomes 2 – so every vertex can have one incoming and one outgoing edge

(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?

33.3.8 Independence

(10 PTS.) 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.

33.3.9 Partition

Text

We already know the following problem is NP-COMPLETE:

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:

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.

33.3.10 Minimum Set Cover

(15 PTS.)

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 PTS.) Prove that MINIMUM SET COVER problem is NP-COMPLETE
(B) (5 PTS.) Prove that the following problem is NP-COMPLETE.

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 PTS.) *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 $p \in \mathcal{I}$.

33.3.11 Bin Packing

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 PTS.) Show that the BIN PACKING problem is NP-COMPLETE
(B) (5 PTS.) Show that the following problem is NP-COMPLETE.

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?

33.3.12 Knapsack

(A) (5 PTS.) Show that the following problem is NP-COMPLETE.

reduction from partition => set s_{i} and v_{i} to be a_{i}

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 $\sum_{u \in U'} v(u) \geq K$.

(B) (5 PTS.) Show that the following problem is NP-COMPLETE.

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
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 PTS.) State this problem as a decision problem.
(ii) (2 PTS.) Show that the decision problem is NP-COMPLETE.

33.3.13 Vertex Cover

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.[2]
(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.

[2] 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.

33.3.14 Bin Packing

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.

TILING

Instance: Finite set $\mathcal{RECTS}$ of rectangles and a rectangle R in the plane.
Question: Is there a way of placing the rectangles of $\mathcal{RECTS}$ 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 $\mathcal{RECTS}$ is a set of squares that can be arranged as to tile R completely. Present a polynomial time algorithm that computes a subset $\mathcal{T} \subseteq \mathcal{RECTS}$, and a tiling of $\mathcal{T}$, so that this tiling of $\mathcal{T}$ covers, say, 10% of the area of R.

33.3.15 Minimum Set Cover

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.

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 $p \in \mathcal{I}$.

33.3.16 k-Center

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.

33.3.17 MAX 3SAT

Consider the Problem MAX SAT.

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.

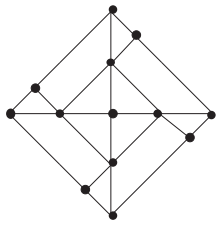


Figure 33.1: Gadget for PLANAR-3-COLOR.

33.3.18 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.

33.3.19 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 $\{-6, 7, -4, -13, -2, 5, 13\}$.

33.3.20 Polynomially equivalent.

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 polynomial time equivalent and which are not? Why?

33.3.21 PLANAR-3-COLOR

Using 3COLORABLE, and the 'gadget' in figure **??**, 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.

33.3.22 DEGREE-4-PLANAR-3-COLOR

Using the previous result, and the 'gadget' in figure **??**, 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.

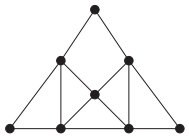


Figure 33.2: Gadget for DEGREE-4-PLANAR-3-COLOR.

33.3.23 Primality and Complexity

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.

33.3.24 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.

33.3.25 Polynomial time Hmiltonian path

(A) Prove that if G is an undirected bipartite graph with an odd number of vertices, then G is non-hamiltonian. 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.

33.3.26 (Really HARD)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.

33.3.27 (t, k)-grids.

(20 pts.)
A graph G is a (t, k)-grid if it vertices are

$$V(G) = \left\{(i, j) \;\middle|\; i = 1, \ldots, n/k, j = 1, \ldots, k\right\},$$

and two vertices (x_1, x_2) and (y_1, y_2) can be connected only if $|x_1 - y_1| + |x_2 - y_2| \leq t$. Here n is the number of vertices of G.

(A) (8 PTS.) Present an efficient algorithm that computes a Vertex Cover of minimum size in a given $(t, 1)$-grid G (here you can assume that t is a constant).

(B) (12 PTS.) Let t and k be two constants.
Provide an algorithm (as fast as possible) that in polynomial time computes the *maximum* size Independent Set for G. What is the running time of your algorithm (explicitly specify the dependency on t and k)?

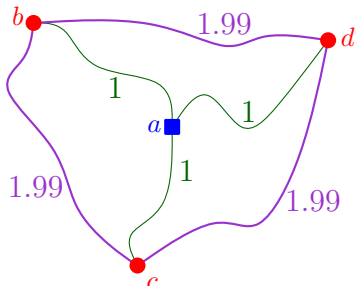
33.3.28 Build the network.

(20 PTS.)

You had decided to immigrate to Norstrilia (never heard of it? Google for it), and you had discovered to your horror that because of import laws the cities of Norstrilia are not even connected by a fast computer network. You join the Roderick company which decided to connect the k major cities by a network. To be as cheap as possible, your network is just going to be a spanning tree of these k cities, but you are allowed to put additional vertices in your network in some other cities. For every pair of cities, you know what is the price of laying a line connecting them. Your task is to compute the cheapest spanning tree for those k cities.

Formally, you are given a complete graph $G = (V, E)$ defined over n vertices. There is a (positive) weight $w(e)$ associated with each edges $e \in E(G)$. Furthermore, you can assume that $\forall i, j, k \in V$ you have $w(ik) \leq w(ij) + w(jk)$ (i.e., the triangle inequality). Finally, you are given a set $X \subseteq V$ of k vertices of G. You need to compute the cheapest tree T, such that $X \subseteq V(T)$, where the price of the tree T is $w(T) = \sum_{e \in E(T)} w(e)$.

To see why this problem is interesting, and inherently different from the minimum spanning tree problem, consider the graph on the right. The optimal solution, if we have to connect the three round vertices (i.e., b, c, d), is by taking the three middle edges ab, ad, ac (total price is 3). The naive solution, would be to take bc and cd, but its cost is 3.98. Note that the triangle inequality holds for the weights in this graph.



(A) (5 PTS.) Provide a $n^{O(k)}$ time algorithm for this problem.

(B) (15 PTS.) Provide an algorithm for this problem with running time $O(f(k) \cdot n^c)$, where $f(k)$ is a function of k, and c is a constant independent of the value of k.

(Comments: This problem is NP-HARD, although a 2-approximation is relatively easy. Problems that have running time like in (B) are referred to as ***fixed parameter tractable***, since their running time is polynomial for a fixed value of the parameters.)