Assignment 1
COMP 4804 - Fall 2017
September 13, 2017

1 Instructions

This is due at the start of the class on the due date. (We will likely be solving these problems in
the class on the due date and hence no late submissions will be accepted.) Please write clearly and
answer questions precisely. As a thumb rule, the answer should be limited to 1 written page/per
Question, with ample spacing between lines and in margins. Always start a new question on a
new page, starting with Question 1, followed by Question 2, ..., Question n. Please cite all the
references (including web-sites, names of friends, etc.) which you have used/consulted as the source
of information for each of the questions. BTW, when a question asks you to design an algorithm -
it requires you to (1) Clearly spell out the steps of your algorithm in pseudocode (2) Prove that
your algorithm is correct and (3) Analyze the running time.

2 Problems

1. Let \( G = (V, E) \) be an undirected connected graph without any cycles, where \( V \) is the set of
vertices and \( E \) is the set of edges. Show that, in polynomial time, you can find a set of vertices
\( V' \subseteq V \) of minimum cardinality, such that for each edge \( e = (uv) \in E \), at least one of \( u \) or \( v \)
is in \( V' \).

2. Let \( G = (V, E) \) be an undirected connected graph. Let \( S \subseteq V \) be the largest subset of vertices
such that there is no edge \( e = (u,v) \in E \) such that \( u,v \in S \). Given \( S \), show that we can
construct a minimum vertex cover of \( G \) in polynomial time. What can you say about the
complexity of finding such a subset \( S \)?

3. Suppose a warehouse in Mississauga packages all the items in boxes that needs to be shipped
to a distributor in Ottawa. For shipping, each box needs to weigh \( \leq W \), where \( W \) is a positive
integer. We can assume that we can fit as many items as we want in a box provided that the
sum total of their weights is at most \( W \). Moreover each item \( I_k, \ k \in \{1, \ldots, n\} \), has weight
\( w_k \), where \( 0 < w_k \leq W \). The cost of shipping is proportional to the number of boxes used.
The strategy the warehouse employs to package these items in boxes is as follows:

(a) Set \( k = 1 \).
(b) Open a new box.
(c) While \( I_k \) can be placed in the open box without exceeding its weight capacity and \( k \leq n \),
   i. Place \( I_k \) in the open box.
ii. Set \( k := k + 1 \).

(d) Close, Tape, and Ship the box.

(e) If \( k \leq n \), GOTO Step b.

Show that the number of boxes used by the warehouse is at most two times the minimum number of boxes required to package all \( n \) items.

4. Let \( x_1, \ldots, x_n \) be \( n \) Boolean variables. We denote by \( \bar{x}_i \) the complement of \( x_i \). A clause consists of OR of Boolean variables and their negations. For example, \((x_3 \lor \bar{x}_7 \lor x_{12})\) is a clause consisting of three literals and it is TRUE if and only if either \( x_3 \) is TRUE, or \( x_7 \) is FALSE, or \( x_{12} \) is TRUE. A 3CNF Boolean formula consists of AND of \( k \) clauses, i.e., \( C_1 \land C_2 \land \cdots \land C_k \) and each clause consists of OR of three boolean literals. Our task is to find out what should be the Boolean values assigned to the variables \( x_1, \ldots, x_n \) so that the maximum number of the clauses are TRUE. Formulate this problem as a 0-1 Integer Linear Program.

5. Suppose you a collection \(|S| + |F|\) parallel machines, where \( S \) is the set of (identical) slow machines and \( F \) is the set of (identical) fast machines. Assume that the fast machines can undertake twice the amount of work as compared to the slow machines per unit of time. We have a set of \( n \) independent jobs, each with its own processing time requirement, that need to be assigned to these machines. Note that if a job requires \( t \) time units for completion, it takes \( t \) time units on a slow machine and \( \frac{t}{2} \) time units on a fast machine. Any job can be performed on any machine, but a job cannot be split into smaller jobs, and once a job is assigned to a machine it cannot be moved to another machine. You need to design an algorithm, running in polynomial time, to assign these jobs to these parallel machines so that the makespan is within three times the optimal makespan. Recall that the makespan of a set of parallel machines is the maximum total processing time, over all the machines, of all the jobs assigned to a machine.

6. (Bonus Problem) Consider the following natural approximation algorithm for the vertex cover problem for a graph \( G = (V,E) \).

\[
\text{Set } C := \emptyset. \text{ Set } E' = E.
\]

(a) Pick a vertex \( v \in V \) of the largest degree in \( G = (V,E') \).

(b) \( C := C \cup \{v\} \).

(c) Remove all edges from \( E' \) incident to \( v \).

(d) Repeat Steps a-c till \( E' \neq \emptyset \).

Return \( C \).

It is easy to see that \( C \) is a vertex cover. You are asked to show that there are graphs \( G \) such that the size of the cover computed by the above method is larger than two times the size of an optimal cover, i.e. \( |C| > 2|C_{OPT}| \), where \( C_{OPT} \) is an optimal cover. Actually, you may try to show that \( |C| = \Omega(\ln n |C_{OPT}|) \). (Hint: To construct such a graph, consider a bipartite graph, where the vertex
set is partitioned in two subsets, say L and R, where \( V = L \cup R \), and all the edges \( e = (u, v) \in E \) satisfy the property that \( u \in L \) and \( v \in R \). You may want to choose vertices in L to have uniform degree, and vertices in R having varying degrees. Recall that \( \sum_{i=1}^{n} \frac{1}{i} = O(\ln n) \).