CS 341, Winter 2015
Andrew Arnold, Timothy Chan, Doug Stinson

Assignment 5 (due **Thursday, April 2, 4pm)**

Please read `https://www.student.cs.uwaterloo.ca/~cs341/policies.html` for general instructions.

1. [*12 marks*] Lieutenant John McClane and shop owner Zeus Carver are by a water fountain and are given two water jugs: one jug that holds n_1 liters of water, the other n_2 liters (where n_1 and n_2 are positive integers). They are allowed to perform the following operations:

   (i) completely fill a jug with water;

   (ii) completely empty a jug of water;

   (iii) pour the contents of one jug into the other jug, stopping only when either the jug being filled is full, or when the jug being emptied is empty.

   Their task is to determine if there is a sequence of such operations that leaves exactly n_3 liters of water in the bigger jug (where n_3 is another positive integer).

   Thankfully, John McClane has taken a third-year algorithms course and realizes that he can pose the problem as a graph problem.

   (a) [*9 marks*] Model the problem as a graph problem. Give a precise definition of the graph involved and state the specific question about this graph that needs to be answered. [*Hint*: define a vertex to be a pair of integers (i, j) where $0 \leq i \leq n_1$ and $0 \leq j \leq n_2$.]

   (b) [*3 marks*] What algorithm can be applied to solve the problem and what is its time complexity as a function of n_1 and n_2?

2. [*12 marks*] Given a set P of n points in 2D, we consider the following *2-clustering problem*: partition P into two subsets P_1, P_2 to maximize

$$d(P_1, P_2) = \min_{p \in P_1} \min_{q \in P_2} d(p, q),$$

   where $d(p, q)$ denotes the Euclidean distance between p and q. (That is, $d(P_1, P_2)$ denotes the smallest distance between P_1 and P_2.)

   (a) [*4 marks*] Let G be the complete graph with vertex set P where the edge pq has weight $w(pq) = d(p, q)$. Let T be the minimum spanning tree of G. Let e be the largest-weight edge in T. Deleting e from T creates two components P_1, P_2. Prove that $d(P_1, P_2) = w(e)$. [*Hint*: you may use a lemma proved in class. You may assume that all edge weights are distinct.]

(b) [*4 marks*] Let e be the edge from part (a). Let P_1^*, P_2^* be the optimal solution. Prove that $d(P_1^*, P_2^*) \leq w(e)$. [*Hint*: again use the lemma from class.]

(c) [*4 marks*] Finally show that the 2-clustering problem can be solved in $O(n^2)$ time by applying an algorithm from class. Use parts (a) and (b) to argue correctness.

3. [*15 marks*]

(a) [*6 marks*] Convert the following optimization problem into a decision problem and show that the corresponding decision problem is in NP.

Input: positive integers $v_1, \ldots, v_n, w_1, \ldots, w_n, W_1, W_2$.
Output: disjoint subsets $A, B \in \{1, \ldots, n\}$ such that $\sum_{k \in A} w_k \leq W_1$ and $\sum_{k \in B} w_k \leq W_2$, while maximizing the total value $\sum_{k \in A} v_k + \sum_{k \in B} v_k$.

[*Note*: You have seen this problem before! Remember that the bit complexity of the input is polynomial in n and $\log U$, where U is the largest integer in the input.]

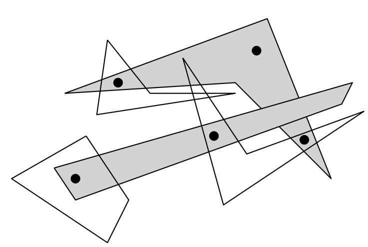
(b) [*3 marks*] Show that if we could solve your decision problem in (a) in time polynomial in the number of input bits, then we could also compute the maximum total value in time polynomial in the number of input bits.

(c) [*6 marks*] Show that if we could solve your decision problem in (a) in time polynomial in the number of input bits, then we could also compute the optimal subsets A and B in time polynomial in the number of input bits. [You may use part (b) as a subroutine.]

4. [*21 marks*] A quadrilateral is a polygon with 4 vertices (which may or may not be convex). Consider the following problem called QUAD-COVER:

Input: a set P of m points, a set Q of n quadrilaterals in 2D, and an integer k.
Output: "yes" iff there exists a subset $T \subseteq Q$ of at most k quadrilaterals such that every point in P lies inside some quadrilateral in T.

In the following example, the answer is "yes" for $k = 2$ (with the optimal subset shaded).



(a) [*4 marks*] Prove that QUAD-COVER is in NP.

Note: you may assume that testing whether a point is inside a quadrilateral can be done in time polynomial in the number of bits.

(b) [*14 marks*] Prove that QUAD-COVER is NP-complete via a polynomial-time reduction (i.e., a polynomial-time transformation) from VERTEX-COVER.

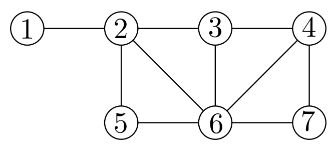
Note: You may use the fact from class that the following problem VERTEX-COVER is NP-complete:

> *Input*: an undirected graph $G = (V, E)$ and an integer k.
>
> *Output*: "yes" iff there exists a subset $S \subseteq V$ of at most k vertices such that for every edge $uv \in E$, we have $u \in S$ or $v \in S$.

More Note: Remember to carefully define your construction of the input to QUAD-COVER given the input to VERTEX-COVER, and remember to prove that your reduction is correct.

Hint: Given a graph $G = (V, E)$ where $V = \{1, \dots, n\}$, to construct the set P of points, map each $ij \in E$ $(i < j)$ to a point (i, j). For the set Q, you will need to define certain non-convex quadrilaterals to cover particular rows/columns somehow...

(c) [*3 marks*] Illustrate your reduction for the following graph G with $k = 3$:



Draw the set P of points and the set Q of quadrilaterals in your construction. Also draw the solution T that corresponds to the vertex cover $S = \{2, 4, 6\}$ of G.