252 COMBINATORIAL OPTIMIZATION

HOMEWORK 1, JANUARY 2014

These exercises are at a mixture of levels. There may be typos or mistakes; please let me know as soon as you find one. I'm happy to discuss your solution attempts in my office hours (Mondays at 12pm).

Hand in Questions 1, 4, 5, 6 and 8 to the box outside the undergraduate office by **Tuesday, 21st January, at 2pm**.

You are *strongly* encouraged to do the other questions at this time as well, though you should not hand them in. You are also encouraged to work on these problems in groups, though your final write-up should be your own.

(1) For each of the following pairs of functions $f, g\colon \mathbb{R} \to \mathbb{R}$ show that $f = O(g)$ as $x \to \infty$ by giving explicit $M > 0$ and x_0 for which $|f(x)| \leq M|g(x)|$ for $x > x_0$.
   (a) $f = 5x^2 - 2x + 7$, $g = x^5$,
   (b) $f = 5x^2 - 2x + 7$, $g = x^2$,
   (c) $f = 5/x$, $g = 1$,
   (d) $f = 3x^2 \log x + 5x \log^2 x$, $g = x^2 \log x$,
   (e) $f = \sum_{i=0}^{k} a_i x^i$, $g = x^k$.

(2) Show that $\log_b(x) = O(\log_a(x))$ as $x \to \infty$ for any $a, b \in \mathbb{R}_{>0}$.

(3) Let $a_1, a_2, \ldots, a_n$ be a list of n integers, which we wish to *sort* into order. For example, if the list is $1, 5, 3$, then the desired output is $1, 3, 5$. For the following two algorithms, check correctness/termination, and calculate the worst-case complexity of each algorithm.
   (a) Insertion sort:
      (i) $i = 1$.
      (ii) While $i \leq n$ do:
         (A) $j = 1$.
         (B) While $a_j \leq a_i$ and $j < i$ do: $j = j + 1$.
         (C) $k = a_j$. $a_j = a_i$. $a_i = k$.
         (D) While $j < i$ do: $j = j + 1$. $k = a_j$. $a_j = a_i$. $a_i = k$.
         (E) $i = i + 1$
      (iii) Output $a_1, \ldots, a_n$.
   (b) Merge sort:

(i) if $n = 0$ or $n = 1$, return $a_1, \dots, a_n$.
(ii) Otherwise Merge sort the list $a_1, \dots, a_{\lfloor n/2 \rfloor}$, and the list $a_{\lfloor n/2 \rfloor + 1}, \dots, a_n$. Output these as $a_1, \dots, a_{\lfloor n/2 \rfloor}$, and $b_1, \dots, b_{\lceil n/2 \rceil}$.
(iii) $i = 1$. $j = 1$. $k = 1$.
(iv) While $i \le \lfloor n/2 \rfloor$ and $j \le \lceil n/2 \rceil$ do:
  (A) if $a_i \le b_j$ then $c_k = a_i$ and $i = i + 1$.
  (B) else $c_k = b_j$ and $j = j + 1$.
  (C) $k = k + 1$.
(v) if $i \le \lfloor n/2 \rfloor$ then while $i \le \lfloor n/2 \rfloor$ do: $c_k = a_i$, $i = i + 1$, $k = k + 1$.
(vi) else while $j \le \lceil n/2 \rceil$ do: $c_k = b_j$, $j = j + 1$, $k = k + 1$.
(vii) Output $c_1, \dots, c_n$.

(c) Bonus: Compare these to other sorting algorithms; for example, quick sort.

(4) Consider the following "algorithm" to find all nonnegative integer solution to the equation $x^3 + y^3 = z^3$.
(a) $i = 1$.
(b) For $j = 1$ to i do
  (i) For $k = 1$ to j do
    (A) if $i^3 + j^3 - k^3 = 0$ then output (i, j, k).
(c) $i = i + 1$.

What is wrong with this algorithm?

(5) Find a MST for the graph of Figure 1.

(6) A computer has a variety of components to be connected by wires. The distance in millimeters between each pair of components is given in the table below. Determine which pairs of components to connect so that the collection of components is connected and the total length of wire between components is minimized.

	1	2	3	4	5	6
1	0	6.7	5.2	2.8	5.6	3.6
2	6.7	0	5.7	7.3	5.1	3.2
3	5.2	5.7	0	3.4	8.4	4.0
4	2.8	7.3	3.4	0	8.0	4.4
5	5.6	5.1	8.4	8.0	0	4.6
6	3.6	3.2	4.0	4.4	4.6	0

(7) Show that the following algorithm finds an MST of a connected graph G. Begin with $H = G$. At each step, find (if one exists) a maximum-cost edge e such that $H \setminus e$ is connected, and delete e from H. Try this algorithm on the graph shown in Figure 1.

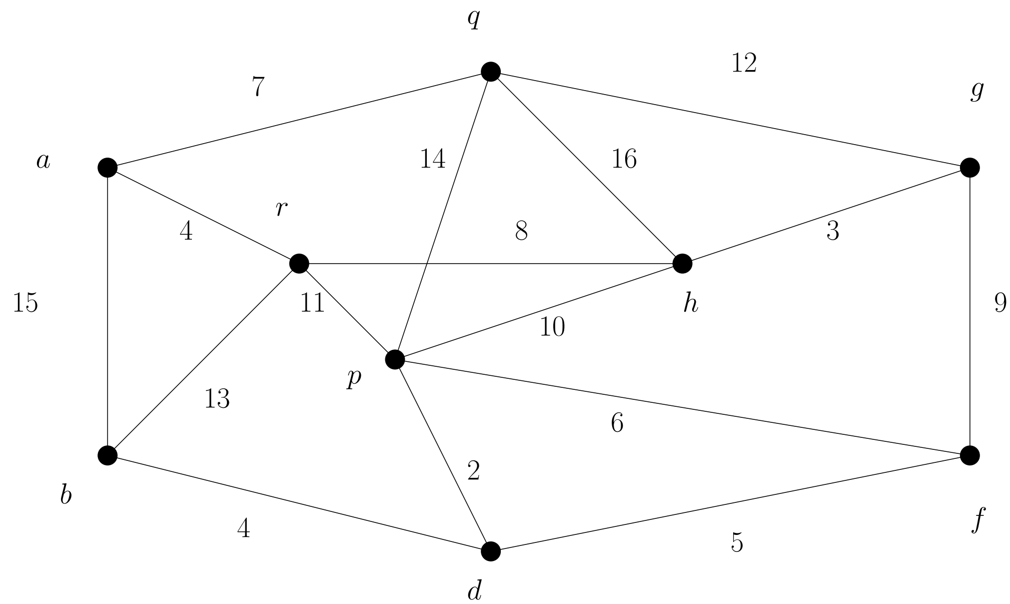

FIGURE 1.

(8) Show that a minimum spanning tree T is unique if and only if any edge (x, y) not in T has larger weight than any edge on the circuit created by adding the edge (x, y) to T.

(9) Consider the quadratic formula algorithm found at: `http://www.cs.amherst.edu/~djv/irs.pdf`. Check its correctness and termination. Challenge: What is the worst-case complexity complexity of this algorithm? (consider this both regarding each operation as one step, and also by taking the bit-length of the input into account). Context: This is a spoof of the US IRS (equivalent of HMRC) forms. Almost all Americans must fill in a tax form every year, so the instructions are at a very low level.

(10) (This is an optional extension exercise for your mathematical culture - it probably requires some Algebra II). In this exercise we show that "Is n prime?" lies in NP, so there is a polynomial-time-checkable certificate for primality. The Wikipedia primality certificate article contains some more information.

(a) Show that if there is $x \in \mathbb{Z}$ with $\gcd(x, n) = 1$, $x^{n-1} \equiv 1 \mod n$, and for any p dividing $n - 1$ $x^{(n-1)/p} \not\equiv 1 \mod n$, then n is prime.

(b) Show that such an x exists for any prime n (harder)

(c) Show that given such an x and a prime factorization of $n-1$, we can check the first condition in $O((\log n)^2)$ multiplications (hint: every integer has $O(\log n)$ prime factors).
(d) This does not suffice for a certificate. We don't want to have to trust the certificate that the prime factorization is correct. For example, what would happen if $n = 85$, and we give $84 = 6 \times 14$?
(e) We solve this problem by giving certificates that each of the prime factors are prime (by giving such an x for each of them). We need to recurse, and give certificates for each of the prime factors other than 2 occuring in each of these certificates. Show that there will be at most $4 \log n - 4$ certificates given in this process.
(f) Conclude that giving such a certificate requires time polynomial in $\log n$.
(g) For more information about the polynomial time algorithm to decide primality, see `www.ams.org/notices/200305/fea-bornemann.pdf`.

(11) (For your mathematical culture). Learn a little more about the P vs NP problem. One place to start is the wikipedia page (`http://en.wikipedia.org/wiki/P_versus_NP_problem#Notable_attempts_at_proof`) on this problem. Some surveys, which start accessible and quickly get hard, are:
(a) Wigderson, "P, NP and mathematics - a computational complexity perspective". (`http://www.math.ias.edu/~avi/PUBLICATIONS/MYPAPERS/W06/w06.pdf`)
(b) Sipser "The history and status of the P vs NP question" (`http://www.eecs.berkeley.edu/~luca/cs172-04/sipser92history.pdf`)
(c) Cook, "The P vs NP problem". (`http://www.claymath.org/millennium/P_vs_NP/Official_Problem_Description.pdf`)

See how far you can get!

(12) (For those who are taking metric spaces) Show that if G is a connected simple graph then the set V with $d : V \times V \to \mathbb{R}$ given by setting $d(v, w)$ to be the distance between v and w is metric space. Give an example to show that this need not be the case for directed graphs.