Algorithms and Data Structures
Spring 2018
Exercises for Units 8 and 9

1. In class we showed how a polynomial f of degree $n-1$ can be simultaneously evaluated at n different values in time $O(n \log^3 n)$. This relied on the fact that if a is a root of a polynomial d then $f(a) = r(a)$, where r is the remainder of f divided by d.

   Show that actually a time bound of $O(n \log^2 n)$ can be achieved.

2. Show that $n!$ can be computed (as a real number) in time $O(\sqrt{n} \log^2 n)$ without the use of floor operations, i.e. no conversions from reals to integers, etc.

   You may assume that a complex N-th primitive root of unity can be obtained in $O(1)$ time for any integer N.

3. In the exercises for a previous unit we considered the so-called "$a + b \neq c$"-problem. Recall that it takes as input three sorted sets A, B, C of $\Theta(n)$ numbers each and asks whether for all $a \in A$, $b \in B$, and $c \in C$ we have $a + b \neq c$.

   For a model of computation that allowed only comparisons (or queries) of the form $x+y < z$, $x + y > z$, or $x + y = z$ with $x \in A$, $y \in B$, and $z \in C$, but no other operations, you proved a lower bound of $\Omega(n^2)$ for the worst case time for solving this problem. This lower bound actually even holds if all the numbers involved are integers from $[0..M]$ with $M = \Theta(n)$.

   Show that if arithmetic operations are allowed and the members of A, B, C are integers from the range $[0..M]$ with $M = \Theta(n)$, then it is possible to solve this problem in $O(n \log n)$ time.

4. You have a set of n items numbered from 1 to n, where item i has weight $w_i > 0$ and all the weights are distinct. You have a truck that can carry a load of at most W. You want to load the truck with as many of the n items as possible while observing this total weight limit W.

   Give an $O(n)$ time algorithm that determines the maximum set of items that can be loaded onto the truck.