Homework set 1

Note: the homework sets are not for submission. They are designed to help you prepare for the quizzes.

1. Consider the following variation of the Interval Scheduling Probem. You have a processor that can operate 24 hours a day, every day. People submit requests to run daily jobs on the processor. Each job j comes with a start time s_j and an end time t_j. If the job is accepted, then it must run during the time interval $[s_j, t_j)$ every day. Notice however that it is possible that s_j occurs before midnight, and t_j after midnight. Given the input set J of jobs, the goal is to accept as many jobs as possible, subject to the constraint that a processor can run at most one job at a time. Give an efficient algorithm to solve this problem. What is the running time of your algorithm?

2. We are given a set J of n jobs. The execution of each job consists of two phases. First, it must be pre-processed on a supercomputer, and then it must be finished on a regular computer. Each job j is associated with parameters $p_1(j)$ - the processing time it requires on the supercomputer, and $p_2(j)$ - the processing time it requires on a regular computer. We only have one supercomputer, which can only execute one job at a time. However, we have an unbounded number of regular computers, that can execute any number of jobs simultaneously. Given a schedule S, the finish time $C(S)$ of the schedule is the earliest time by which all the jobs have been completed. Our goal is to find a schedule S of all jobs, with minimum finish time $C(S)$. Design an efficient algorithm, prove its correctness, and analyze the running time.

3. Suppose we have an alphabet with 2^k characters, and a string in which all characters are almost equally common. That is, for all $x, y \in \Sigma$, $f(x) \leq f(y) < 2f(x)$. How will the Huffman tree look like? What is its cost? Prove your answer.

4. Given a string X, we denote by $X[i]$ the ith character of X. Given an n-character string A, and two additional strings B and C, we say that string A is an *interleaving* of strings B and C iff we can partition the set $\{1, \ldots, n\}$ of indices into two disjoint subsets $I = \{i_1, i_2, \ldots, i_k\}$ and $J = \{j_1, j_2, \ldots, j_{n-k}\}$, where $i_1 \leq i_2 \leq \cdots \leq i_k$ and $j_1 \leq j_2 \leq \cdots \leq j_{n-k}$ such that:

   - $I \cup J = \{1, \ldots, n\}$
   - the string $(A[i_1], A[i_2], \ldots, A[i_k]) = B$, and the string $(A[j_1], A[j_2], \ldots, A[j_{n-k}]) = C$

   In other words, A is obtained by interleaving the characters of B and C. Design an efficient algorithm, that, given as input strings A, B and C, decides whether A is an interleaving of B and C. Prove the algorithm's correctness and analyze its running time.