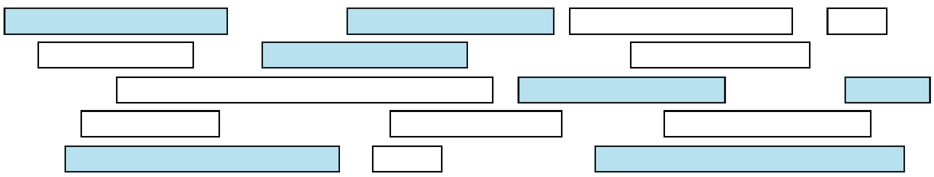
Problem Sheet [1]

1. Describe an efficient algorithm that, given a set $\{x_1, x_2, \ldots, x_n\}$ of points on the real line, determines the minimum number of unit-length closed intervals that contains all of the given points. Prove the correctness of your algorithm.

2. Consider the following scheduling problem: n jobs are given as input. Job j ($1 \leq j \leq n$) has a processing time p_j ($p_j > 0$) and a non-negative weight w_j ($w_j \geq 0$). We must construct a schedule for these jobs on a single machine such that at most one job is processed at each point in time, and each job must be processed non-preemptively; that is, once a job begins to be processed, it must be processed completely before any other job begins its processing. The objective is to find a schedule that minimizes the weighted sum of completion times: $\sum_{j=1}^{n} w_j C_j$. Suppose that jobs are indexed such that $\frac{w_1}{p_1} \geq \frac{w_2}{p_2} \geq \ldots \geq \frac{w_j}{p_j} \geq \ldots \geq \frac{w_n}{p_n}$. Then prove that it is optimal to schedule the jobs in the order $(1, 2, \ldots, j, \ldots, n)$ (job 1 first, job 2 second, and so on).

3. There exists an $O(n)$-time deterministic algorithm (M) for finding median of n given numbers. Using this algorithm as a subroutine, design an $O(n)$-time deterministic algorithm for solving the fractional knapsack problem (items are $(I_i)_{i=1}^{n}$, weight of items are $(w_i)_{i=1}^{n}$, profit of items are $(p_i)_{i=1}^{n}$, and knapsack capacity is W), and also prove the correctness and its time complexity.

4. Suppose we are given a set I of n intervals on the real line. We define a subset of intervals $J \subseteq I$ covers I if the union of all intervals in J is equal to the union of all intervals in I. The size of a cover is the number of intervals. Describe an efficient algorithm to compute the smallest cover of I. Use greedy algorithm, and prove the correctness of your algorithm.



A set of intervals, with a cover (shaded) of size 7.

5. **To reach n from 1:** Let learn a new process to reach from 1 to n. Initially start with 1, and in each step you can either increment by 1 or double the integer. Your goal is to reach n. For example to reach from 1 to 10,

$$1$$

$$1 + 1 = 2$$

$$2 * 2 = 4$$

$$4 + 1 = 5$$

$$5 * 2 = 10$$

In 4 steps, we can reach from 1 to 10. Describe and analyze an algorithm to compute the minimum number of steps required to produce any given integer n from 1 only using two strategies, i.e. incrementing by 1 or doubling.

6. Let C be a unit radius circle. An arc of C is given by a pair $[\theta_1, \theta_2]$, where $\theta_1 < \theta_2$ are angles between 0 and 360 degrees. You are given a set of n arcs in the circle and would like to select a subset of arcs of maximum cardinality so that no two of them overlap. Give an efficient algorithm to find an optimal solution.

[1] Prepared by Pawan K. Mishra