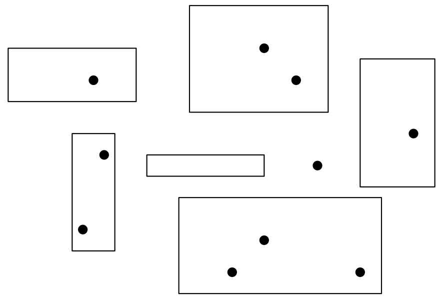
CS 473, Spring 2023
Homework 1 (due Feb 1 Wed 9pm)

Instructions: Carefully read `http://courses.grainger.illinois.edu/cs473/sp2023/policies.html#hw` and `http://courses.grainger.illinois.edu/cs473/sp2023/integrity.html`.

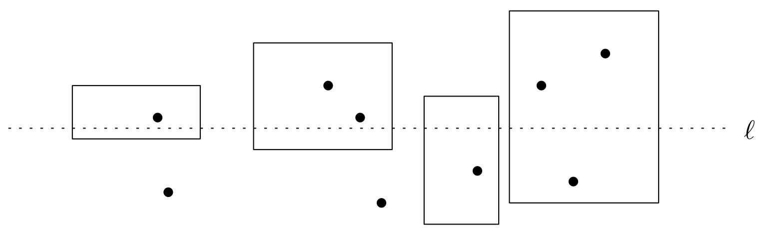
- **Groups of up to three people may submit joint solutions.** Each problem should be submitted by exactly one person, and the beginning of the homework should clearly state the Gradescope names and email addresses of each group member. In addition, whoever submits the homework must tell Gradescope who their other group members are.
- **Submit your solutions electronically on the course Gradescope site as PDF files.** Submit a separate PDF file for each numbered problem. If you plan to typeset your solutions, please use the LaTeX solution template on the course web site. If you must submit scanned handwritten solutions, please use a black pen on blank white paper and a high-quality scanner app (or an actual scanner, not just a phone camera).
- If you are not using your real name and your `illinois.edu` email address on Gradescope, you will need to fill in the form (see course website).
- **You may use any source at your disposal**—paper, electronic, or human—but you *must* cite *every* source that you use, and you *must* write everything yourself in your own words.
- Any homework or exam solution that breaks any of the following rules may be given an *automatic zero.*
  - Always give complete solutions, not just examples.
  - Always declare all your variables, in English. In particular, always describe the specific problem your algorithm is supposed to solve.
  - Always avoid unnecessarily long answers.

Problem 1.1: We are given a set R of n_1 rectangles in 2D, where the sides of the rectangles are parallel to the x- and y-axes (i.e., the rectangles are not rotated) and the rectangles do not overlap. We are also given a set P of n_2 points in 2D. For every point $p \in P$, we want to find the rectangle $r_p \in R$ that contains p. (Note that if p is not covered by any of the rectangles in R, then r_p is undefined, but otherwise r_p is unique because of the nonoverlapping assumption.) Let $n = n_1 + n_2$.



(a) (35 pts) First give an $O(n \log n)$-time algorithm for the special case of the problem where all rectangles of R are assumed to intersect a given horizontal line ℓ.

[Hint: sort...]



(b) (65 pts) Now describe an algorithm to solve the general problem in $O(n \log^2 n)$ time or better.

[Hint: use divide-and-conquer and part (a) as a subroutine.][1]

Problem 1.2: Given n integers $a_1, \ldots, a_n, b_1, \ldots, b_n$, define the function

$$F(x) = \frac{1}{a_1 x + b_1} + \frac{1}{a_2 x + b_2} + \cdots + \frac{1}{a_n x + b_n}.$$

$F(x)$ is a *rational function*, i.e., a ratio of two polynomials (of degree at most n). Describe an efficient algorithm to compute the coefficients of these polynomials. You may assume that arithmetic operations on numbers take constant time (even though the coefficients could be very large). The running time should be of the form $O(n \log^c n)$ for some constant c.

[1] As usual, in a multi-part question, if you are unable to solve (a), you can still do (b) under the assumption that (a) has been solved.

For example:

$$\frac{1}{x+2} + \frac{1}{3x-1} + \frac{1}{2x+1} \;=\; \frac{11x^2 + 11x - 1}{6x^3 + 13x^2 + x - 2}.$$

[*Hint*: divide-and-conquer. Use polynomial multiplication, i.e., convolution, as a subroutine.]

Problem 1.3: We are given a text string $T = t_1 t_2 \cdots t_n \in \Sigma^*$ of length n over alphabet Σ. We are also given a sequence of pairs $P = \langle (a_1, b_1), \ldots, (a_m, b_m) \rangle$ with $a_i, b_i \in \Sigma$. We say that there is a *match* iff there exists a position i such that for every $j \in \{1, \ldots, m\}$, we have $t_{i+j} = a_j$ or $t_{i+j} = b_j$. Describe an efficient algorithm to determine whether there is a match.

For example: for $T = 01\mathbf{230}1032$ and $P = \langle (1,2), (0,3), (0,1) \rangle$ with $\Sigma = \{0,1,2,3\}$, there is a match (at the position shown in bold).

[*Hint*: use convolution.]

[*Note*: A correct solution with $O(|\Sigma| n \log n)$ run time would get full credit, i.e., 100 pts. A correct solution with $O(n \log n)$ run time (even when $|\Sigma|$ is large) would get 5 bonus pts.]