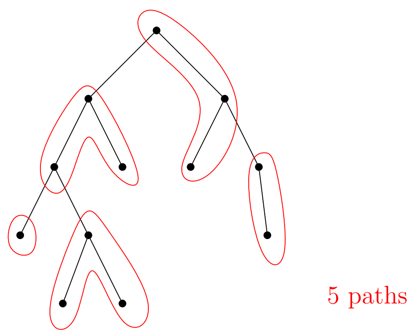
CS/ECE 374 A (Spring 2024)
Homework 7 (due Mar 21 Thursday at 10am)

Instructions: As in previous homeworks.

Problem 7.1: Given a binary tree T with n nodes, we want to find a collection of paths, each with 1, 2, or 3 nodes, such that every node is in exactly one path, while minimizing the number of paths.



Design and analyze an efficient dynamic programming algorithm for computing the minimum number of paths. Include the following steps: (i) first define your subproblems precisely, (ii) then derive the recursive formula (including base cases) with brief justifications, (iii) specify a valid evaluation order, and (iv) analyze the running time. For this problem, you do not need to write pseudocode if your recursive formula and evaluation order are described sufficiently clearly. And you do not need to output the optimal collection of paths.

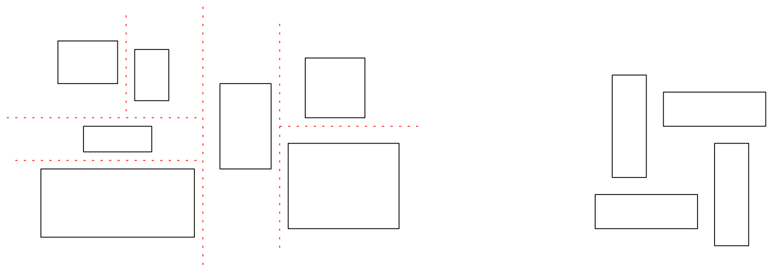
(Hint: the example on independent sets in trees that will be covered in 3/7 Thursday's lecture might be helpful. See also Problem Old.7.1 in Past HW7 for another similar example.)

Problem 7.2: An (axis-parallel) rectangle in 2D can be represented by four real numbers: the min and max x-coordinates and the min and max y-coordinates.

We define the notion of a *perfectly separable* set of rectangles inductively as follows:

- A set with just one rectangle is perfectly separable.
- If S_1 is a perfectly separable set and S_2 is a perfectly separable set and there is a vertical line ℓ such that all rectangles of S_1 are completely to the left of ℓ and all rectangles of S_2 are completely to the right of ℓ, then $S_1 \cup S_2$ is perfectly separable.
- If S_1 is a perfectly separable set and S_2 is a perfectly separable set and there is a horizontal line ℓ such that all rectangles of S_1 are completely below ℓ and all rectangles of S_2 are completely above ℓ, then $S_1 \cup S_2$ is perfectly separable.
- Only sets satisfying the above are perfectly separable.

In the examples below, (i) is perfectly separable, but (ii) is not.

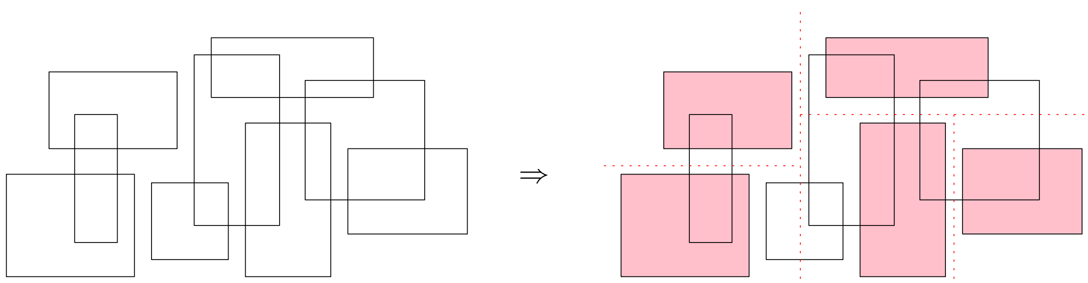


(i) (ii)

(Remark: It has been conjectured that any set of n non-overlapping rectangles contains a subset of at least n/c rectangles that is perfectly separable, for some constant c. This conjecture is still open. . .)

Consider the following problem: given a set S of n (possibly overlapping) rectangles,[1] find a largest subset $T \subseteq S$ such that T is perfectly separable.

Design and analyze a polynomial-time dynamic programming algorithm to compute the optimal size. Include the following steps: (i) first define your subproblems precisely, (ii) then derive the recursive formula (including base cases) with brief justifications, (iii) specify a valid evaluation order, and (iv) analyze the running time. For this problem, you do not need to write pseudocode if your recursive formula and evaluation order are described sufficiently clearly. And you do not need to output the optimal subset.



5 rectangles

(Hint: create $O(n^4)$ subproblems: for each x, x', y, y', define a subproblem restricted to input rectangles that are inside $[x, x'] \times [y, y'] \ldots$)

(Another Hint: As you can see from the pictures above and the recursive definition of a perfectly separable set, a feasible solution has a binary-tree-like structure where each node corresponds to a vertical or horizontal cut. The example in Lab 8b and Problem Old.7.1 in Past HW7 also involve a binary-tree-like structure, and so might be helpful.)

[1]You may assume that all x- and y-coordinates are distinct.