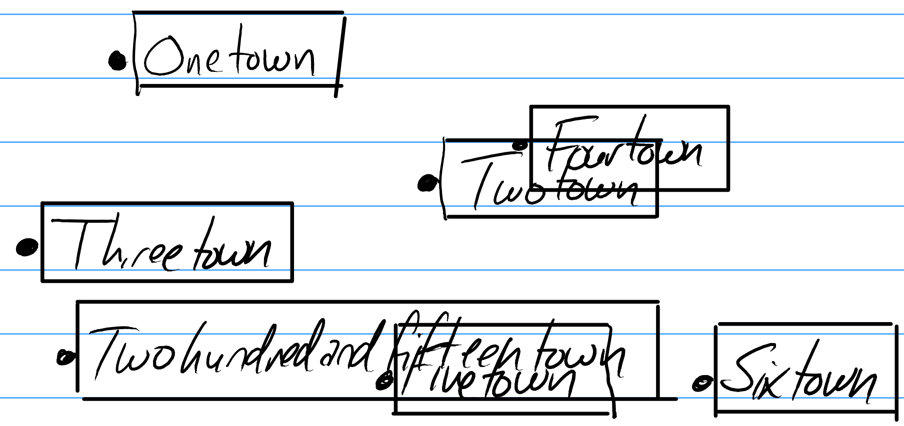
Map Labelling with Fixed-Height labels
(Maximum Independent Set in Unit Height Rectangles)

P.K. Agarwal, M. van Kreveld, S. Suri. Label placement by maximum independent set in rectangles. Computational Geometry: Theory and Applications. vol. 11. pp. 209–218, 1998.



Input: A set R of n rectangles, all of which have height 1.

Output: A subset $R' \subseteq R$ of maximum cardinality, and such that no two elements of R' intersect each other.

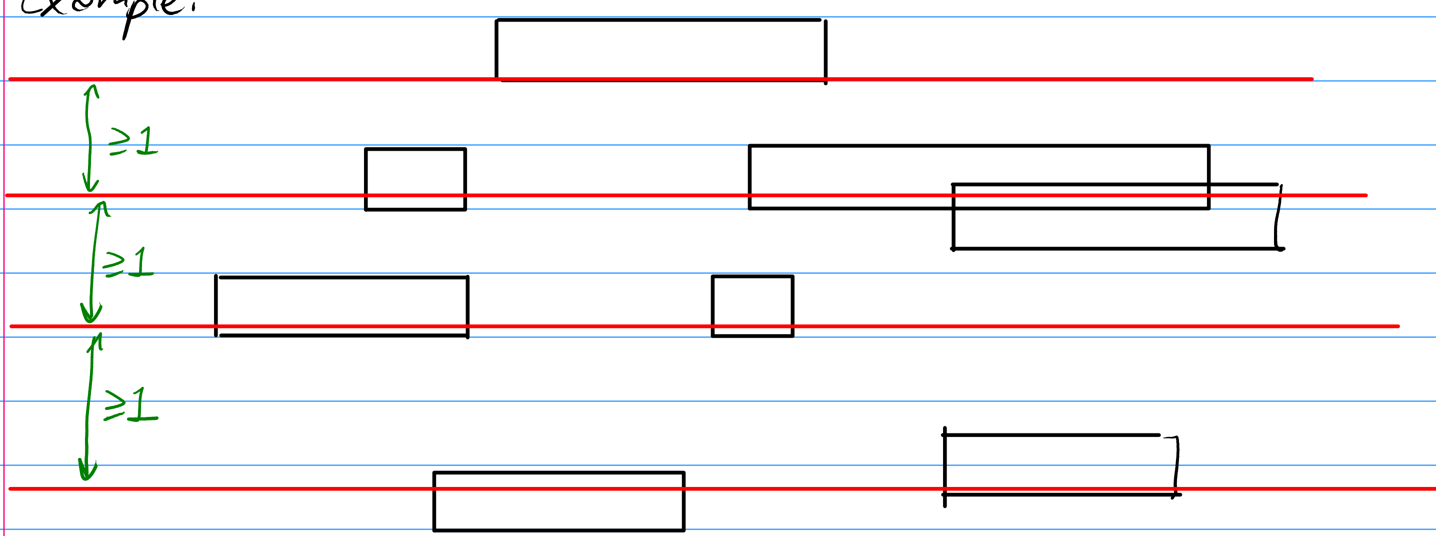
Note: This problem is NP-hard because it generalizes the problem of Maximum Independent Set among unit squares.

First-Up: A 2-Approximation

1. Draw a set of horizontal lines $L_1, \dots, L_K$ such that.

   (a) The distance between any two lines is at least 1.
   (b) Each line intersects at least on rectangle in R.
   (c) Each rectangle in R intersects at least one line.

Example:



Explained later. ⇒ 2. Find an MIS R_i' of the rectangles that intersect L_i, for each $i \in \{1, \dots, K\}$

3. Output $R_1' \cup R_3' \cup R_5' \dots R_K''$ or $R_2' \cup R_4' \cup \dots \cup R_{K-1}'$, whichever is bigger (for odd K).

Claim: The algorithm above outputs a $\frac{1}{2}$-approximation to optimal solution.

Proof: Let R^* denote an optimal solution.

Among the rectangles of R that intersect L_i, R^* can not include more than $|R_i'|$ of these. Therefore,

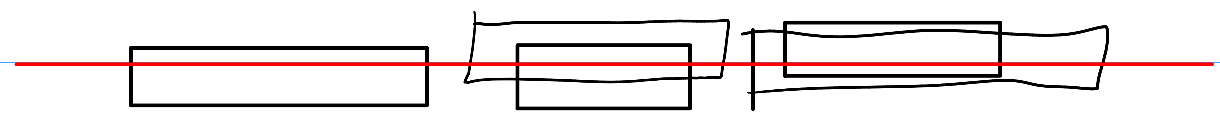
$$|R^*| \leq |R_1'| + |R_2'| + \cdots + |R_k'| .$$

But the algorithm outputs a set of rectangles of size at least

$$\frac{1}{2}\left(|R_1'| + |R_2'| + \cdots |R_k'| \right),$$

so the algorithm is a $\frac{1}{2}$-approximation $\blacksquare$

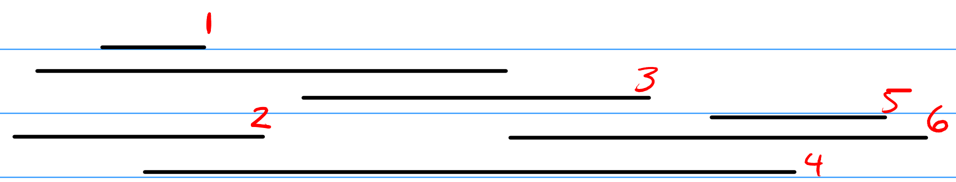
How to find the MIS of rectangles that all intersect L_i?



Greedy Algorithm (for rectangles that all intersect L_i)

Sort rectangles by their right side, then repeatedly add a rectangle to R_i' as long as it doesn't interfere with any rectangle that's already there.

Example:



- Add 1
- Don't add 2 (because it interferes with 1)
- Add 3
- Don't add 4 (it overlaps 2)
- Add 5
- Don't add 6. (it overlaps 5)

Theorem: The greedy algorithm correctly outputs a maximum independent set of rectangles and takes $O(n \log n)$ time.

This is everything we need for:

Theorem: There exists an $O(n \log n)$ time algorithm that finds a $\frac{1}{2}$-approximation for MIS is unit-height rectangles.

A $(1-1/t)$-Approximation

Idea: Same as before, but instead of solving 1 line at a time, we will solve $t-1$ lines at a time

- Let R_i be the subset of R intersected by L_i
- Let $R_i^t = R_i \cup R_{i+1} \cup \cdots \cup R_{i+t-1}$
- We make t subproblem $S_1, \ldots, S_t$. In each subproblem we delete one out of every t lines
- Let $R_i = \emptyset$ if $i < 1$ or $i > K$.
- Then $S_j = R_{j-t} \cup R_j \cup R_{j+t} \cup R_{j+2t} \cup \cdots$ for each $j \in \{1, \ldots, t\}$.

- Algorithm: Find the optimal solution for each S_j and output the maximum.

Claim: The above algorithm outputs a $(1-1/t)$-approximation.

Proof: Let R^* be an optimal solution and let
$R^*_i = R^* \cap R_i$.

Let $OPT(S)$ denote an optimal solution for set S.

Observe that

$$|OPT(S_j)| \geq |R^*| - |R^*_{j-1}| - |R^*_{j-1+t}| - |R^*_{j-1+2t}| - \cdots$$

Therefore: $$\sum_{j=1}^{t} |OPT(S_j)| \geq t \cdot |R^*| - |R^*_1| - |R^*_2| - |R^*_3| - \cdots |R^*_k|$$

$$= (t-1)|R^*|.$$

But then $\max\{|OPT(S_j)| : j \in \{1,\ldots,t\}\} \geq (k-1)|R^*|/t$

$$= \left(1-\frac{1}{t}\right) \cdot |R^*|. \quad \blacksquare$$

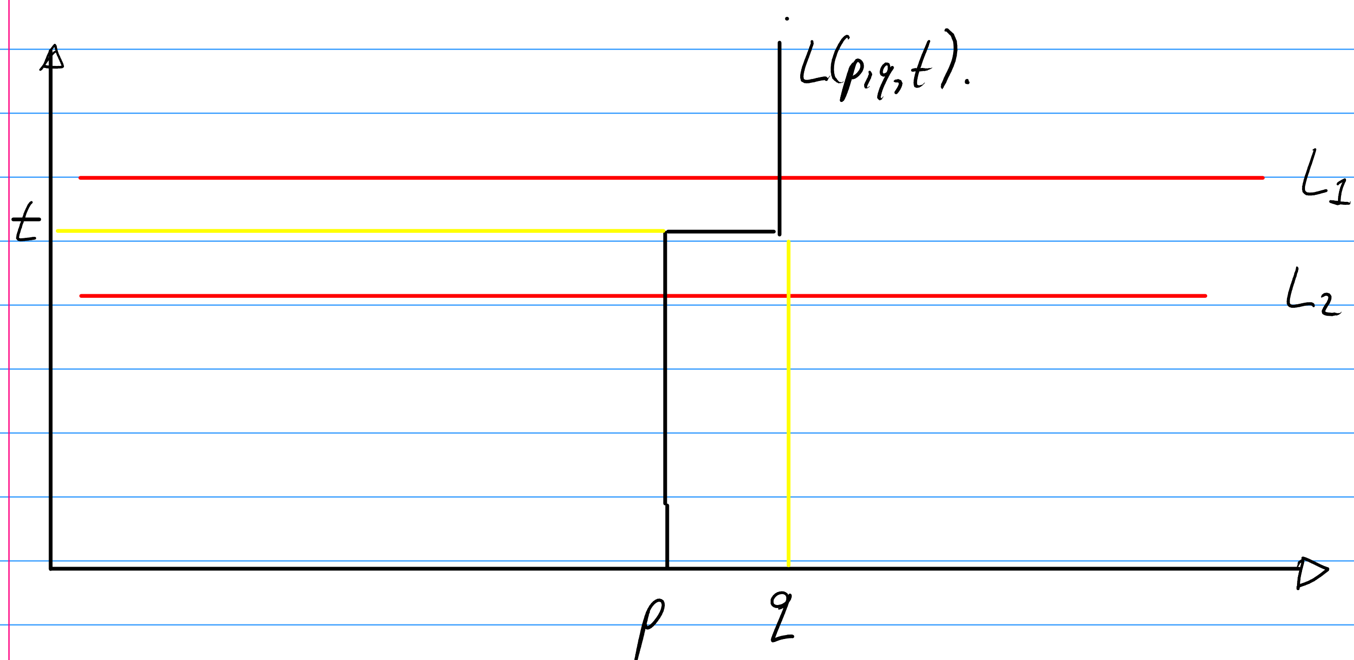
All that remains is to show how to find the MIS of $R_i, \ldots, R_{i+t-1}$.

Dynamic Programming:

For simplicity, consider only two lines, say L_1 & L_2.

We solve a sequence of subproblems parametrized by 3 values, p, q, and t.

Subproblem (p, q, t) finds the optimal solution using only rectangles that lie to the left of the polyline $L(p, q, t)$



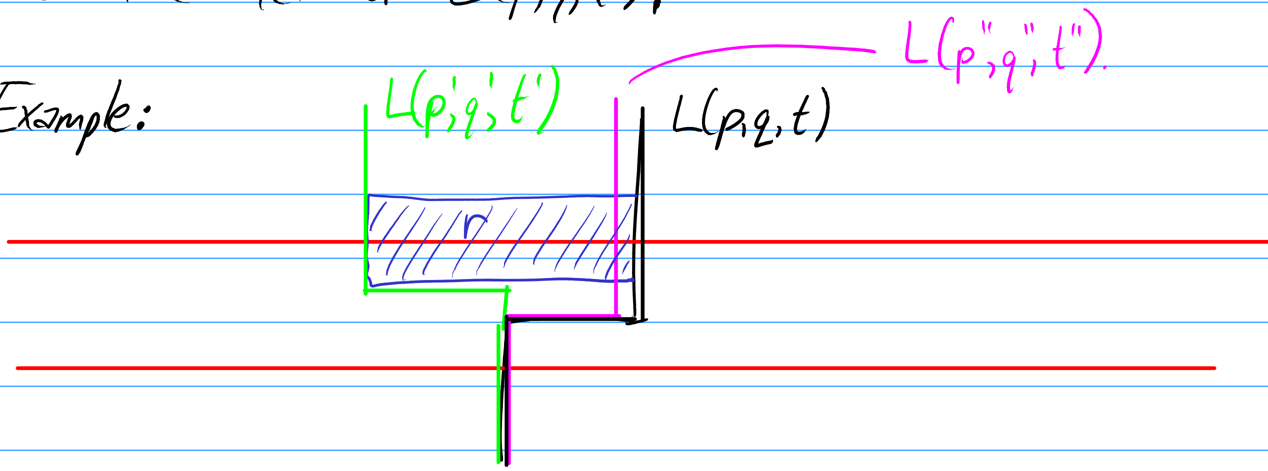
How many values of p, q, t do we have to worry about?

- one value of p for every vertical edge in R_1
- one value of q for every vertical edge in R_2
- one value of t for every top edge in R_2 and every bottom edge in R_1

- $2|R_1| \cdot 2|R_2| \cdot (|R_1| + |R_2|) = O((|R_1| + |R_2|)^3)$

Claim: We can easily compute solution to p, q, t if we know all solutions (p', q', t') we $L(p', q', t')$ is completely to the left of $L(p, q, t)$.

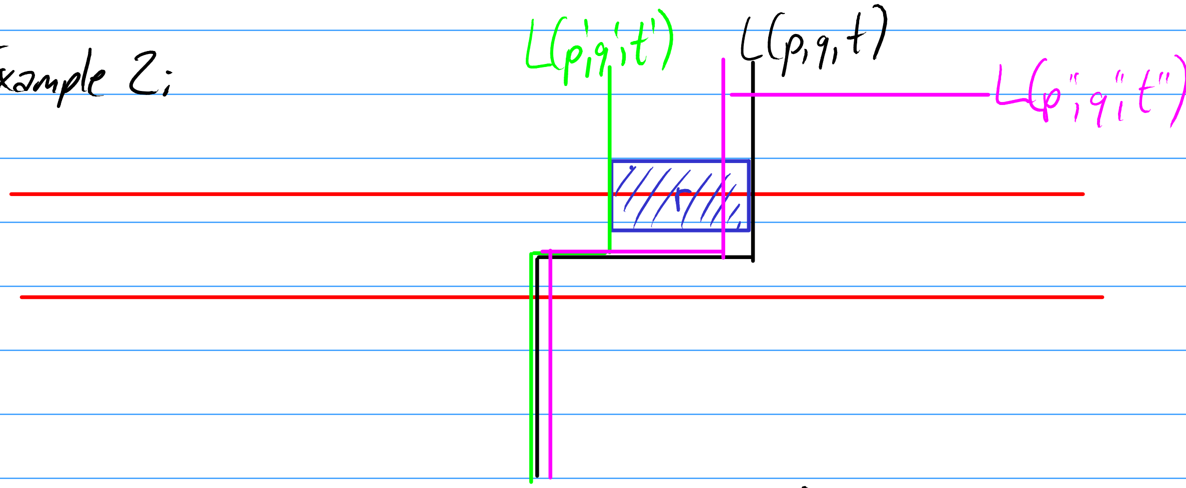
Example:



Optimal solution either includes r or it doesn't.

- Includes r: $1 +$ solution for $L(p', q', t')$.
- Doesn't include r: solution for $L(p'', q'', t'')$.

Example 2:



Includes r: $1 +$ solution for $L(p', q', t')$
Doesn't include r: solution for $L(p'', q'', t'')$.

- Cases involving rectangles with right edge on R_2 are similar to the above examples.

- Using this method we can compute the solutions to all problems in $O((|R_1|+|R_2|)^3)$ time.

<u>Theorem:</u> There exists an $O(n^3)$ time algorithm that gives a $(1-\frac{1}{3})$-approximation for MIS in unit height rectangles.

The above theorem generalizes. By taking groups of size $t-1$, we obtain

<u>Theorem:</u> There exists an $O(n^{2t-1})$ time algorithm that gives a $(1-1/t)$-approximation for MIS in unit-height rectangles.