1 December 2021

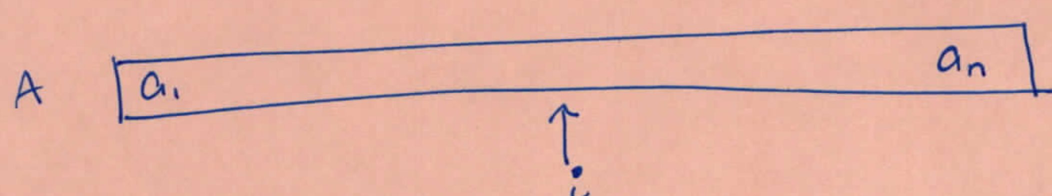
RQS(A, s, t)
Base case!

$\boxed{\begin{array}{l} i \leftarrow \text{a rand. int btwn } s \,\&\, t\text{, inclusive} \\ \text{pivot } A \text{ on } A[i] \\ \text{RQS on 1}^{\text{st}} \text{ "half"} \leftarrow \text{recur. call} \\ \text{RQS on 2}^{\text{nd}} \text{ "half"} \leftarrow \text{recur. call} \end{array}}$

Worst-case RT : $T(n) = \Theta(n) + T(n-1)$
$= \Theta(n^2)$

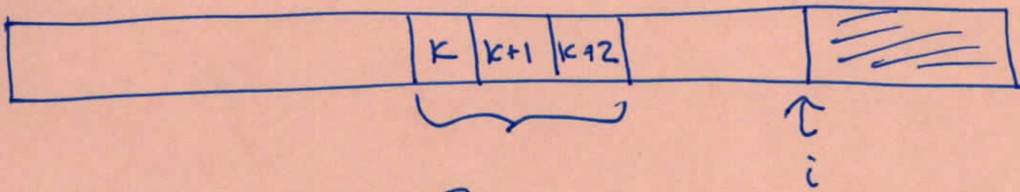
the runtime boils down to counting comparisons
Let's suppose A, sorted is:

$a_1, a_2, \ldots, a_n$
(assume no duplicate values)



Q: What is the prob. that a_j and a_k are compared?
- the pivot gets compared w/ everyone, so if $j = i$ or $k = i$, then they are compared
- if $k < i$ and $j > i$, not compared.
- if k, j are on the same side, we're not sure yet.

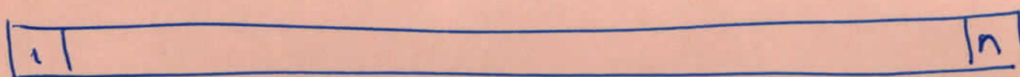
- if $j = k+1$, they must eventually be compared
- if $j = k+2$,



$$\mathbb{P}(C_{k,k+2}) = \frac{2}{3 = k-j+1}$$

↑ the event that a_k and a_{k+2} are compared

- if $j=1$, $k=n$



$$\mathbb{P}(C_{1,n}) = \frac{2}{n = k-j+1}$$

- in general, $\mathbb{P}(C_{j,k}) = \frac{2}{k-j+1}$ w/ $j \le k$

So, what is the expected # of comparisons?

$$\mathbb{E}(\#\text{comparisons}) = \sum_{j=1}^{n} \sum_{k=j+1}^{n} \mathbb{E}[C_{j,k}]$$

# of times compared / value of comparison

$$= \sum_{j=1}^{n} \sum_{k=j+1}^{n} \mathbb{P}[C_{j,k}] \cdot 1$$

$$= \sum_{j=1}^{n} \sum_{k=j+1}^{n} \frac{2}{k-j+1}$$

$$= \sum_{j=1}^{n} 2\left(\frac{1}{n-j+1} + \frac{1}{n-j} + \dots + \frac{1}{1}\right) = \Theta(n \log n)$$

How many comparisons btwn elts of A?

$$\mathbb{E}(\#\text{ comparisons}) = \mathbb{E}\left(\sum_{j=1}^{n}\sum_{k=j+1}^{n} n_{j,k}\right)$$

\# times j, k are compared = 1 if comp, 0 if not.

$$= \sum\sum \mathbb{E}(n_{i,j}) \text{, by lin. of expect.}$$

$$= \sum\sum \mathbb{P}[C_{j,k}]\cdot 1 + \cancel{\mathbb{P}[\sim C_{j,k}]\cdot 0}$$

$$= \sum\sum \mathbb{P}[C_{j,k}]$$

<u>Note</u>: Without the randomization, we can't do expected case analysis. Our adversary could send us "bad" input each time.

<u>Note</u>: We <u>could</u> have "bad" runtime, but the probability of that is so low!! (as long as we have that randomized step).

Linear Programming

the problem is stated as follows:

- goal: minimize (or maximize)

$$\boxed{\sum_{i=1}^{n} a_i x_i}$$

a linear objective function

variables

constants

subject to:

$$\begin{array}{ll} \text{ineq. constraints} & \left\{ \begin{array}{l} \sum_{i=1}^{n} b_i x_i \geq c_1 \\ \sum_{i=1}^{n} d_i x_i \leq c_2 \end{array} \right. \\ \text{eq. constraints} & \left\{ \sum_{i=1}^{n} \alpha_i x_i = \beta \right. \end{array}$$

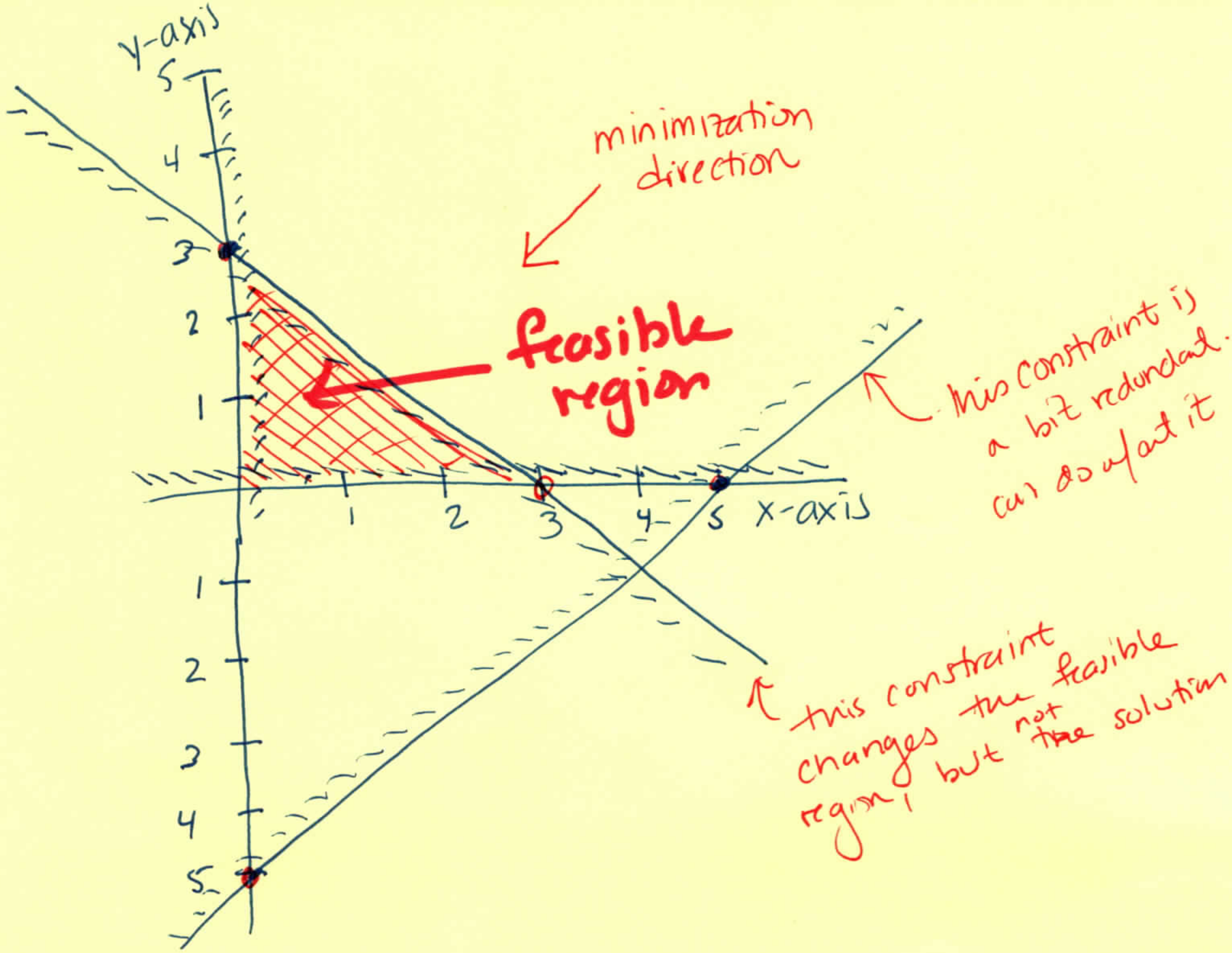
→ linear constraints. can have any # of them!

e.g., minimize $\overbrace{ax + by}^{x+y}$, $a=1, b=1$

subject to $\left.\begin{array}{l} x \geq 0 \\ y \geq 0 \end{array}\right\}$ often constraints!

$$x + y \leq 3$$
$$x - y \leq 5$$

in groups: solve by Drawing in ~~the~~ $\mathbb{R}^2$.



- in $\mathbb{R}^3$, our lines become planes
- in $\mathbb{R}^4$ & higher, lines become hyperplanes
- the region is minimized either at a corner (generically defined by 2 lines) or on a line (a bit degenerate of a situation)

  unique sol'n

  ↳ non-unique sol'n
- note: our feasible region can be empty, in which case there is no sol'n:

e.g.,

Standard Form of LP

Any LP can be turned into standard form.

maximize $\sum_{i=1}^{n} a_i x_i$

subject to $\sum_{i=1}^{n} b_{ij} x_i \leq c_j$ for $j = 1, \ldots, k$

and

$x_i \geq 0$

- to turn min. prob to max: just mult. the objective fn by -1.
- to turn $\geq$ constraint to $\leq$, mult. both sides by -1.
- to turn $=$ constraint to $\leq$, make two inequalities $a=b \Leftrightarrow a \leq b$ and $b \leq a$