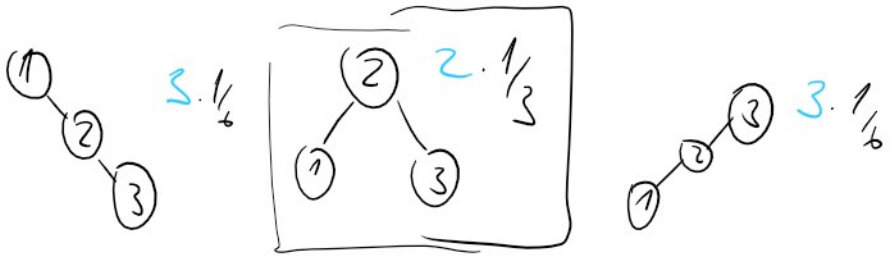
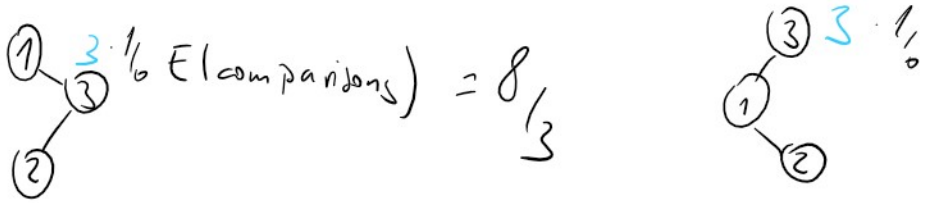
Tutorial 2

17 February 2022 14:44





$f(n) > E(\text{comparisons w. input size } n)$

Expected number of steps is $O(n \cdot \log n)$ where $|S| = n$

S - set of the element to order $\{1, \dots, n\}$

$\forall i, j \in S$ define a random variable S_{ij}

$S_{ij} = 1$ iff i and j are compared in a run

$S_{ij} = 0$ iff i and j are not compared during the run

$$X = \sum_{i<j} S_{ij}$$

$$E(X) = E\left(\sum_{i<j} S_{ij}\right) = \sum_{i<j} E(S_{ij}) = \sum_{i<j} Pr(S_{ij} = 1)$$

$$E(S_{ij}) = 0 \cdot Pr(S_{ij} = 0) + 1 \cdot Pr(S_{ij} = 1)$$

1.) What is probability to compare i and j in a single run of RQUICKSORT ?

1.) What is probability to compare i and j in a single run of RQUICKSORT ?

$$Pr(\text{i and j get compared in 1st step}) = \frac{2}{|S|}$$

Step 2 $|S_1|$, y, $|S_2|$

what is the probability to compare i and j in step 2?

$$\begin{cases} Pr(\text{i and j get compared} \mid i < y < j) = 0 \\ Pr(\text{i and j get compared} \mid y < i < j) = \frac{2}{|S_2|} \\ Pr(\text{i and j get compared} \mid i < j < y) = \frac{2}{|S_1|} \end{cases}$$

(3) → $i \in |S_1|$, $j \in |S_2|$

Step k: → Let's bound the probability of i and j in step k if both i and j are in the same subset $|S_k|$

$$Pr\left(\text{i and j compared in step k} \;\middle|\; \text{i and j are in same subset after k rounds}\right)$$

$$|S_k| \geq j - i + 1$$

$\underbrace{i, i+1, \ldots, j}_{(j-i+1)}$

$$Pr(\text{i and j compared in step k} \mid \text{they are both in } S_k) = \frac{2}{|S_k|} \leq \frac{2}{j-i+1}$$

$S_{ij} = 1$ iff i and j got compared in some round.

$$Pr(S_{ij}) = Pr(\text{i and j get compared in round k} \mid \text{they are both in } |S_k|)$$

$$Pr(S_{ij}) = Pr(i \text{ and } j \text{ get compared in round } k \mid \text{they are both in } |S_k|)$$

$$\cdot Pr(\text{they are in the same set})$$

$$+ \underbrace{Pr(i \text{ and } j \text{ get compared in round } k \mid \text{they are not in same set})}_{=0}$$

$$\cdot Pr(\text{they are not in the same set})$$

$$\geq Pr(i \text{ and } j \text{ get compared in round } k \mid \text{they are both in } |S_k|)$$

$$\cdot Pr(\text{they are in the same set})$$

$$\leq Pr(i \text{ and } j \text{ get compared in round } k \mid \text{they are both in } |S_k|)$$

$$\leq \frac{2}{j-i+1}$$

$$E(X) \leq \sum_{i<j} \frac{2}{j-i+1} = \dots = n \cdot \sum_{i=1}^{n} \frac{1}{i} = \Theta(n \log n)$$

Classification of randomized algorithms.

- Complexity classes
- Algorithm types

Complexity classes

DTM - deterministic TM

$a \in \Sigma$ ← initial type

$|a|$

$p \in Q$

$$(Q \times \Sigma) \to (Q \times \{\leftarrow, \rightarrow, \downarrow\})$$

$Acc \subseteq Q$

Decision problems: $x \in \Sigma^*$ (input belongs to a language $L \subseteq \Sigma^*$

↓ a set of all words over alphabet Σ

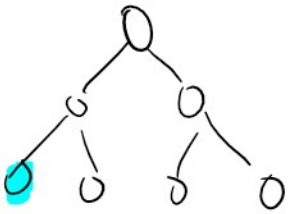
TM ends in an accepting state for $x \in L$

TM ends in a rejecting state for $x \notin L$

NTM $(Q \times \Sigma) \times (Q \times \{\in, \rightarrow, \downarrow\})$

(multiple evolution rules)

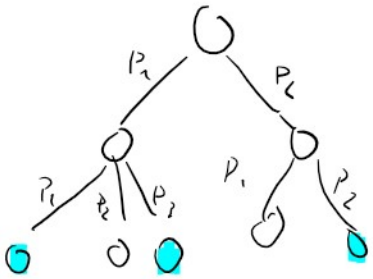
$x \in L \Rightarrow \exists$ accepting state



Probabilistic TMs

→ multiple rules for some (state, input) but there are assigned probabilities

Accepting states have define a probabilities



RP → random polynomial

contains problems for which there is a PTM

s.t.

run in poly time s.t.

$x \in L : Pr(TM(x) \text{ accepts}) \geq 1/2 \geq \varepsilon > 0$

$x \notin L : Pr(TM(x) \text{ accepts}) = 0$

Problems in RP have 1-sided MC algorithm with NO-bias

co-RP containing problems s.t. $\exists TM$

$x \in L : Pr(TM(x) \text{ accepts}) = 1$

$x \notin L : Pr(TM(x) \text{ accepts}) \leq 1/2 \leq \varepsilon < 1$

They have 1-sided MC algorithms with YES-bias

Probability amplification

We want to show that a small number (polynomial in input size) of repetitions of 1-sided MC algorithms is sufficient to decrease the probability of error to an arbitrary small number.

Run the algorithm k times.

1.) How do you choose the answer (in RP = NO bias)
if there is a YES answer the answer of repetition is YES and it is correct.

2.) IFF k NO answers are obtained the answer is NO.
What is the probability of error?

If for the original algorithm the probability of error is ε,

If for the original algorithm the probability of error is ε_1

then the repetition error is $\varepsilon = \varepsilon_1^k$.

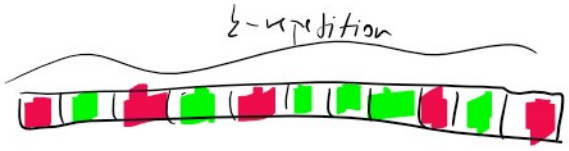
BPP - bounded probabilistic polynomial

$x \in L$: $Pr(TM(x) \text{ accept}) \geq 3/4$ $> 1/2$

$x \notin L$: $Pr(TM(x) \text{ accepts}) < 1/4$ $< 1/2$

2-sided error MC algorithm

Again small number of repetitions can achieve arbitrarily small number of errors.



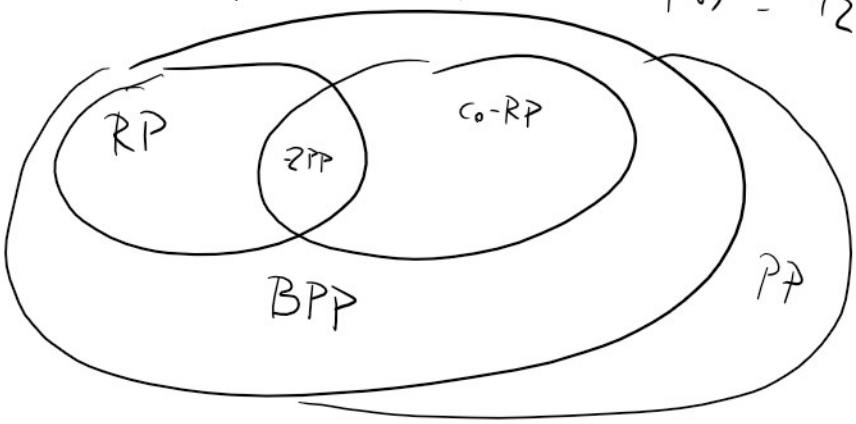
→ majority voting more answers win.

Chernoff-bounds (NEXT tutorial)

PP - probabilistic polynomial

$x \in L$: $P_r(TM(x) \text{ accepts}) > 1/2$ $(1/2 + 1/2^n)$

$x \notin L$: $Pr(TM(x) \text{ accepts}) \leq 1/2$



ZPP → associated with LAS VEGAS

ZPP → associated with LAS VEGAS algorithms

0.) ZPP = co-RP ∩ RP ⇒ problems have both 1-sided MC with YES bias and NO bias

1.) LV algorithm of type 1: Always gives a correct answer and has **expected running time** (in input size) in poly

2.) LV algorithm of type 2: Runs in poly-time but can give an indefinite '/ don't know' answer. w.p. $\leq 1/2$

1.) ⇒ 2.) Run type 1 algorithm and if it runs for too long (!?) say '/ don't know'.

$2E(n)$

$$Pr(X \geq 2E(X)) \leq \frac{1}{2}$$

2.) ⇒ 1.) if type 2 algorithm says '/ don't know', run it again

Define r.v. X which is equal to number of repetitions

$E(X)$ will be polynomial.

0.) => 1) We have A_y - gives correct YES answer in polytime with probability $\geq 1/2$

A_N - $\ldots$

Run $A_y(x)$ and if it gives a correct answer 'YES' output it. Otherwise run $A_N(x)$ and if it says 'NO' it is a correct answer otherwise

2.) => 1.)

→ Run $LV_2(x)$ if a correct answer is found give it as an output.

If received 'I don't know' answer

→ $A_y(x) \rightarrow NO$

→ $A_N(x) \rightarrow YES$