2/25/21

Lower bounds

P = NP ? 1970 ≈ 50 years

3SAT Formula

$$\bigwedge_{i=1}^{m} (x_i \vee \bar{x}_j \vee x_k) \quad x_1, \ldots, x_n \in \{0,1\}$$

Q: Is there a satisfying assignment?

$m 2^n$

Fastest running time

SETH ~ strong exponential time hypothesis

KSAT can not be solved in time faster than 2^n if k is sufficiently large

---

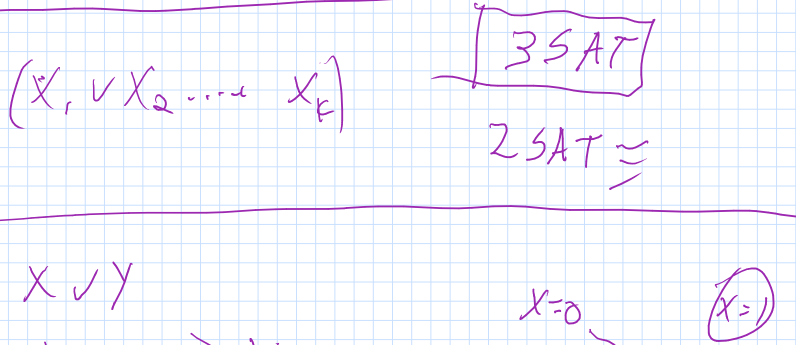
$(x_1 \vee x_2 \ldots x_k)$ — 3SAT

2SAT ≅

$x \vee y$

$x = 0 \Rightarrow y = 1$

$y = 0 \Rightarrow x = 1$



$2^{\Omega(n)}$ 3SAT

---

polynomial — exponential time

$n^{O(1)}$ |———| 2^n

$2^{O(\log n)}$ $2^{O(\sqrt{n})}$ $2^{O(n)}$

$2^{\log^{O(1)} n}$

Sorting

$e_1, e_2, \ldots, e_n$ distinct

Only compare elements

$$comp(i,j) = \begin{cases} 0 & e_i > e_j \\ 1 & e_i < e_j \end{cases}$$

Comparision model

Thm

Any deterministic algorithm performs $\Omega(n \log n)$ comparisions when sorting n numbers in the comparision model.

2SAT

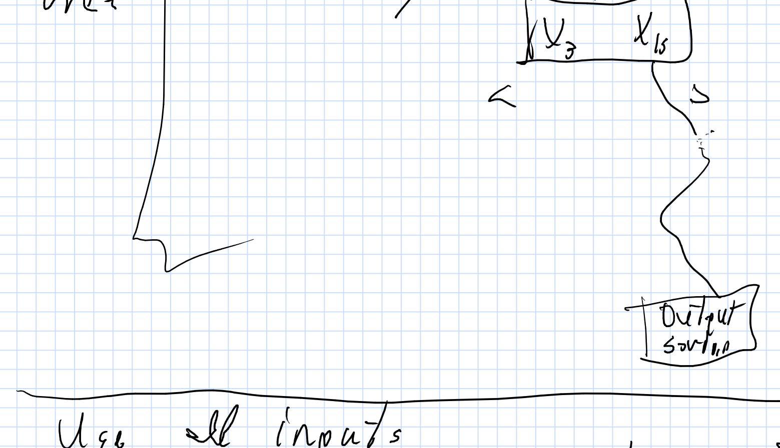
$x_1 = 0, \ldots x_n = 0$

$(x_i \vee x_j)$ $O(n^2)$

↑ ↑ + $\sqrt{t}$

3SAT

Deterministic algorithm



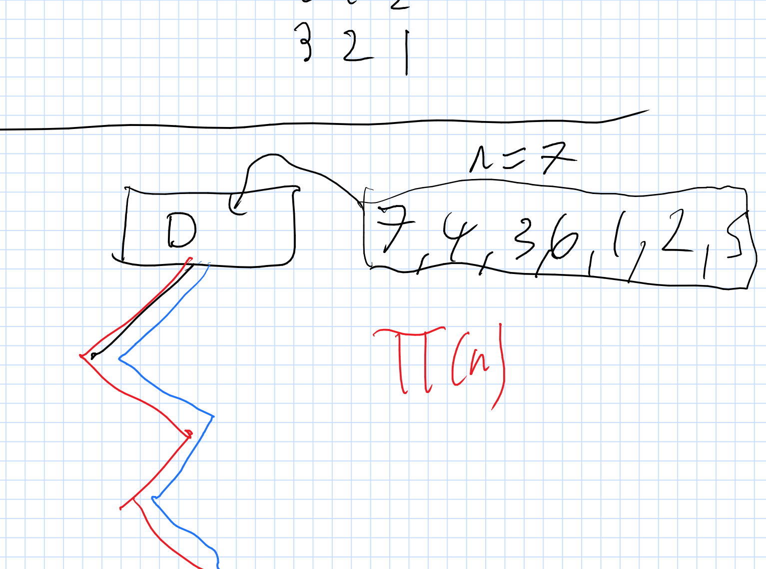
Use all inputs

$\Pi(n) = \{$ all permutations of $1, 2, \ldots n \}$

$|\Pi(n)| = n!$

1, 2, 3
1 3 2
2 1 3
2 3 1
3 1 2
3 2 1

3!



$n!$ different inputs

Every input ends up in a different leaf

⟹ The decision tree has $n!$ leafs.

~ Binary tree

– A tree of height h has at most 2^h leaves.

⟹ $2^h \geq n!$ h = height of the decision tree

$$\left(\frac{n}{2}\right)^{\frac{n}{2}} \leq n! \leq n^n$$

$$h = \lg 2^h \geq \ln n! \geq \lg \left(\frac{n}{2}\right)^{n/2} = \frac{n}{2} \lg \frac{n}{2}$$ ∎

Uniqueness

$e_1, \ldots e_n$ elements

$$comp(e_i, e_j) = \begin{cases} -1 & e_i < e_j \\ 0 & e_i = e_j \\ +1 & e_i > e_j \end{cases}$$

Q: Are all the numbers in the input unique

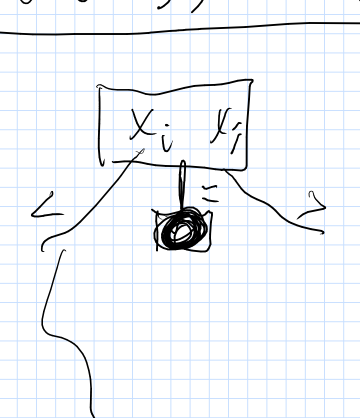
A: Sort + Scan. $O(n \log n)$

Thm

In the comparision model UNIQ requires $\Omega(n \log n)$ time.

---

proof



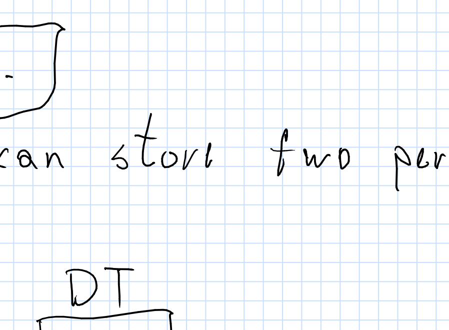
$n!$

$\Pi(n)$ all permutations $1, \ldots, n$

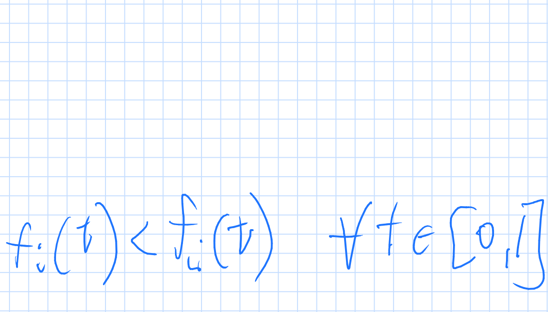
Lemma No leaf can store two permutations

proof

$\pi, \sigma \in \Pi(n)$

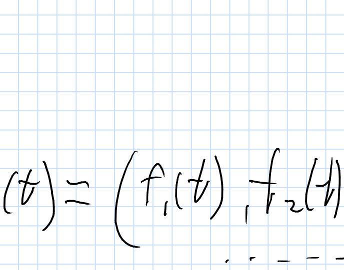


$f_i(t) < f_j(t) \quad \forall t \in [0,1]$



$$F(t) = (f_1(t), f_2(t), \ldots f_n(t))$$

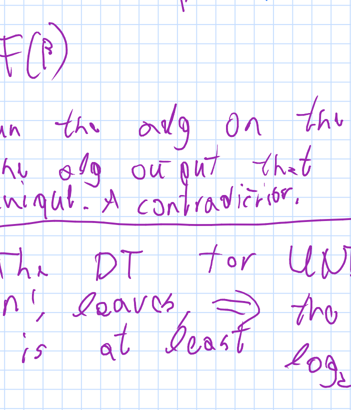
$$f_i(t) = t\pi(i) + (1-t)\sigma(i) \quad t \in [0,1]$$



$F(t) = (f_1(t), f_2(t), \ldots, f_n(t))$

$F(0) = \pi$ $F(1) = \sigma$

$\exists i, j$ $\pi(i) < \pi(j)$, $\sigma(i) > \sigma(j)$



$F(\beta)$

Run the alg on the input $F(\beta)$. The alg output that all numbers are unique. A contradiction. ∎

The DT for UNIQ must have $n!$ leaves ⟹ the height of tree is at least $\log_2 n! = \Omega(n \log n)$.

---

closest pair in linear time

B

3SUM

$A, B, C \subseteq \mathbb{R}$ $|A| = |B| = |C|$

Q: $\exists a \in A, b \in B, c \in C$ s.t. $a + b = c$.

$O(n^3)$ super naive

$O($ A, B, C sort them

$a_1\ a_2\ a_3 \ldots a_n$

$b_1\ b_2 \ldots b_n$

$c_1\ c_2 \ldots c_n$

For $i = 1$ to n do

$a_i + b_1, a_i + b_2, a_i + b_3 \ldots a_i + b_n$

$c_1 < c_2 < c_3 \ldots < c_n$

do merge $O(n)$

if two equal numbers then bingo.

$O(n^2 + n \log n) = O(n^2)$ ⇐

$X = \{a_i + b_j \mid i \in [n], j \in [n]\}$

sort X $O(n^2 \log n)$

$O(n^2 \log n)$

Fine grained complexity