HONORS ALGORITHMS

2024-01-10

ADAPTIVE DECISION TREE

↑

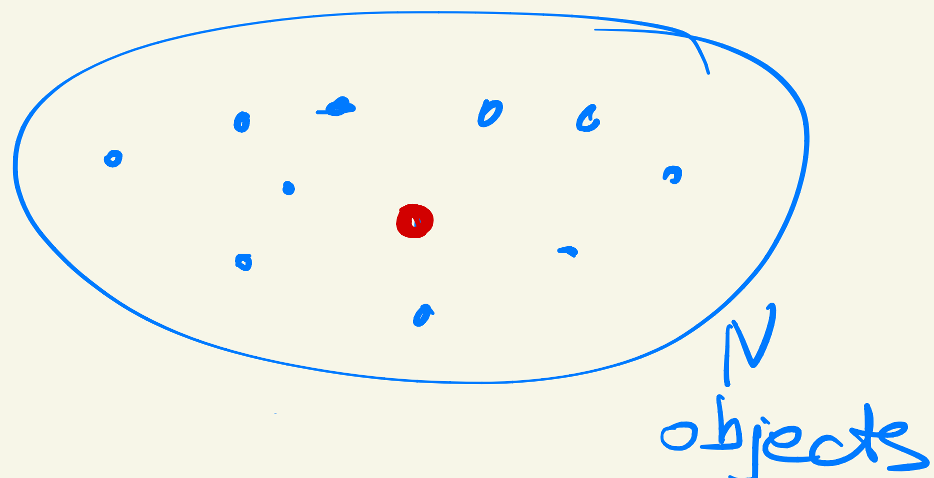
Questions depend on previous answers

vs. OBLIVIOUS

DEPTH-k decision tree

max k steps from root to leaf

\# Y/N sequences of length k : 2^k

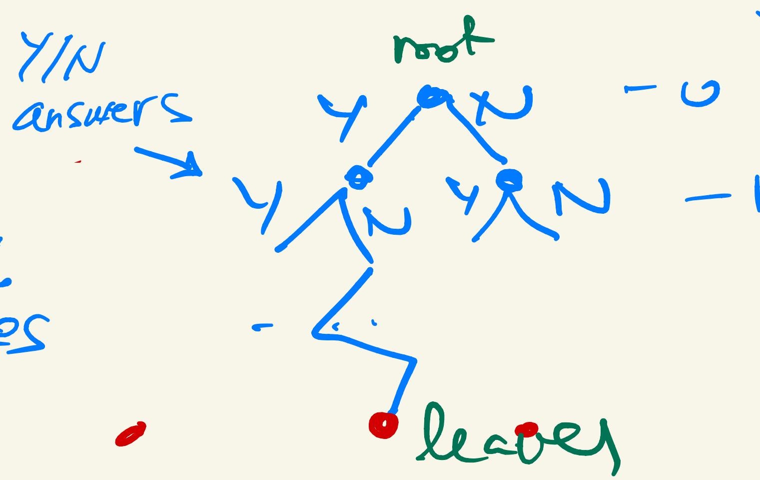


to separate all the N possible outcomes

we need $2^k \geq N$

$k \geq \lceil \log_2 N \rceil$ ← for binary decision trees

$\boxed{\text{information theory lower bound}}$



Comparison-based sorting

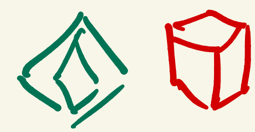
cost : # comparisons

free : overhead

$\begin{cases} x > y \\ x \le y \end{cases}$

Sort crystals by hardness

comparison: scratching



n objects

outcomes : $n!$

$\therefore$ # comparisons $\geq \log_2(n!) \sim n \log_2 n$

possible comparisons $\binom{n}{2} = \frac{n(n-1)}{2}$

MERGE-SORT

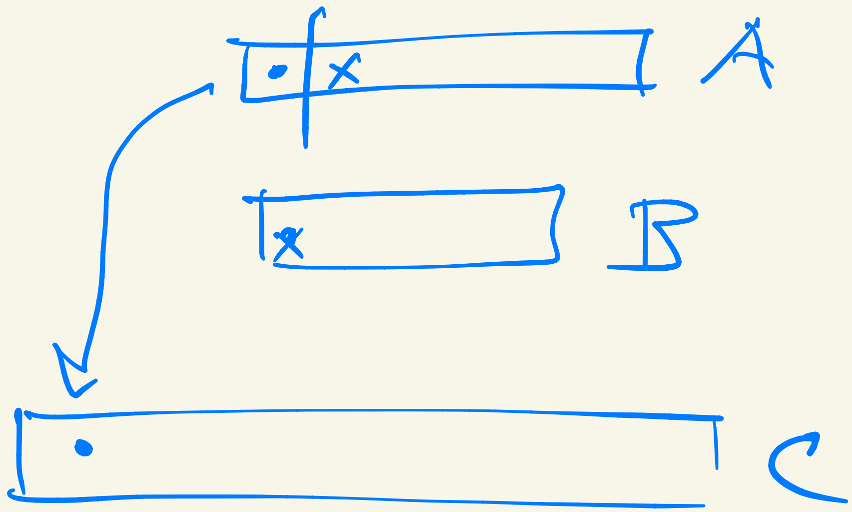
$A[x_1, \ldots x_m]$ $\quad B[y_1 \ldots y_n]$ $\quad x_i, y_i \in \mathbb{R}$

sorted lists: $x_1 \le x_2 \le \ldots \le x_m$ $\quad y_1 \le \ldots \le y_n$

Output: $C[z_1, \ldots z_{m+n}] \longleftarrow$ merged list

suff: # comparisons:

$m+n-1$



MERGESORT

$\lfloor \frac{n}{2} \rfloor + \lceil \frac{n}{2} \rceil = n$

if $n \geq 2$

$$\text{SORT } A[1 \ldots n] =$$

$$\text{MERGE}\Big(\text{SORT } A[\underbrace{1 \ldots \lfloor \tfrac{n}{2} \rfloor}_{\lfloor \frac{n}{2} \rfloor}],\ \text{SORT } A(\underbrace{\lfloor \tfrac{n}{2} \rfloor + 1, \ldots, n}_{\lceil \frac{n}{2} \rceil})\Big)$$

floor

ceiling

$T(n)$: # comparisons used

$$T(n) \leq T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + T\left(\left\lceil \frac{n}{2} \right\rceil\right) + n - 1$$

$T(1) = 0$

$T(2) = 1$

$T(3) \leq 3$

$T(4) \leq 5$

$T(5) \leq 8$

upper bound

$T(3) \leq T(1) + T(2) + 3 - 1 = 3$

$T(4) \leq T(2) + T(2) + 4 - 1 = 5$

$T(5) \leq T(2) + T(3) + 5 - 1 = 8$

lower bound

n	$\lceil \log_2(n!) \rceil$
1	0
2	1
3	3
4	5
5	7

Assume for simplicity $n = 2^k$ ~~WLOG~~ ← losing a factor of 2 when extending to non-powers of 2

$$T(n) \le 2T\left(\frac{n}{2}\right) + n - 1$$

method of reverse inequalities:

guess $g(n)$ s.t.

$$g(n) \ge 2g\left(\frac{n}{2}\right) + n - 1 \qquad g(1) \ge T(1)$$

$$\therefore \quad (\forall n)(g(n) \ge T(n))$$

$$g(n) = An \cdot \log_2 n + Bn + C$$

6

$$g(n) = A n \log_2 n + Bn + C$$

need: $g(n) \geq 2g\left(\frac{n}{2}\right) + n - 1$, $g(1) \geq 0$

wish $A = 1$

$$A n \log_2 n + Bn + C \overset{?}{\geq} 2A \cdot \frac{n}{2}(\log_2 n - 1) + 2 \cdot B \cdot \frac{n}{2} + 2C + (n-1)$$

$$-C \geq -An + (n-1)$$

$$An - C \geq (n-1)$$

$A := 1$, $C := 1$ ✓

$g(1) \overset{?}{\geq} 0$

$$A \cdot 1 \cdot \underbrace{\log_2 1}_{0} + B \cdot 1 + C \overset{?}{\geq} 0$$

$$B + C \geq 0$$

$$B \geq -1$$

$B := -1$ ✓

$\therefore (\forall k)\left(n = 2^k \Rightarrow T(n) \leq n \log_2 - n + 1\right)$ $\sim n \log_2 n$

$\boxed{\text{HW}}$ $(\forall n)(T(n) \le n \log_2 n)$

AMORTIZED COST OF INCREMENTING

<u>for</u> $k = 0$ <u>to</u> $2^n - 1$

DO something

<u>endfor</u>

010111 + 1

1000

$\boxed{\text{HW}}$ average cost of increments = ?