COP 3502 10/8/25

Exam Location : MSB-260
Time: 10am - 12pm (have room until 12:20pm)
Date: 10/11/2025 Saturday!!!

Spr 2022

Best Case O(1) on 1st pass of while,
if is false, else if true → return

Worst Case O(lgn) → jump starts at n/2,
repeatedly divides by 2 similar to binary search
so # iterations ≤ O(lgn)

Sum 2020

Ask increasing powers of 2 until you get no
Jump the largest pow of 2 for which you got
yes, then ~~ask~~ ask smaller powers of 2.
decrementing by 1 each time + jumping for each
yes.
Each jump you make in second ~~half~~ part will
be a unique power of 2, the sum of which
will equal n(n-1).
Run time is no worse than $2 \times \log_2 n = O(\lg n)$
1 log factor increasing jumps, plus log decreasing

n boxes the first k have prizes the last $n-k$ boxes don't. Want to determine exact value of k.

Questions: What's the contents of box i?

Must give our answer after 2^{nd} no response.

What's the fewest # of questions we need to ask to guarantee w/ solve the problem correctly?

Once you get your 1^{st} no, you must sequentially ask about

box x, box $x+1$, box $x+2$, ...

where box $x-1$ was ~~no~~ your max yes.

Q: k, $2k$, $3k$, $\frac{n}{k}-k$ $\frac{n}{k}$

$\#Qs = \frac{n}{k}$ (phase 1)

$\#Qs = k-1$ (phase 2)

$$\min \boxed{\frac{n}{k} + k} - 1$$

Let $k = n^c$

$$\min \frac{n}{n^c} + n^c - 1 = \boxed{n^{1-c} + n^c}$$

What value of c minimize $\max(1-c, c)$, $c \in [0,1]$

$c = \frac{1}{2}$

Strategy: $\sqrt{n}, 2\sqrt{n}, 3\sqrt{n}, \ldots \quad n \qquad \max \sqrt{n}$

$$\left.\begin{array}{l} a\sqrt{n} = \text{yes} \\ (a+1)\sqrt{n} = \text{no}. \end{array}\right\} \quad \left.\begin{array}{l} a\sqrt{n}+1 \\ a\sqrt{n}+2 \\ \vdots \\ a\sqrt{n}+(\sqrt{n}-1) \end{array}\right\} \sqrt{n}-1$$

Run-time $O(\sqrt{n})$

Towers of Hanoi

Towers (n, s, e)

Towers (n-1, s, mid)
Move
Towers (n-1, mid, e)

Closed form like $n^3 + 2n$ no reference to $T(n)$ on RHS... or Sum.

Let $T(n)$ = # moves required to solve the towers puzzle w/ n disks

Recurrence Relation

$$T(n) = T(n-1) + 1 + T(n-1), \quad T(1) = 1$$

initial condition/ base case

$$T(n) = \underline{2T(n-1) + 1}$$
$$= 2\big(2T(n-2) + 1\big) + 1$$
$$= 4T(n-2) + (2+1)$$
$$= \underline{4T(n-2) + 3}$$
$$= 4\big(2T(n-3) + 1\big) + 3$$
$$= 8T(n-3) + 4 + 3$$
$$= \underline{8T(n-3) + 7}$$

$T(n-1) = 2T(n-2) + 1$

$T(n-2) = 2T(n-3) + 1$

After k iterations, we find

$$T(n) = 2^k \underline{T(n-k)} + (2^k - 1),$$

We know $T(1)$ we want $n-k=1$, $k = n-1$.

Plug in $k = n-1$

$$T(n) = 2^{n-1}\,T(n-(n-1)) + (2^{n-1}-1)$$

$$= 2^{n-1}\,T(1) + 2^{n-1} - 1$$

$$= 2^{n-1} + 2^{n-1} - 1$$

$$= \boxed{2^n - 1}$$

$2^{n-1}(1+1) - 1$

$2^{n-1}(2^1) - 1$

$2^{n-1+1} - 1$

$2^n - 1$

reason