Information

Thanos Search

You are good friends with *Thanos*, an intergalactic super-villain appearing in children's entertainment franchises.

Thanos is willing to help you with your searching algorithm. For a sequence `A` and key `k`, a call to `thanos(A, k)` will remove from `A` exactly half of the entries, but not `k`. (We assume that `k` appears at most once in `A` to make this well-defined.) To fix notation, let's agree that `A` has length `n`. The time for a call to `thanos` is $T(n)$; the function modifies `A` in place, so after

```
A = [1,5,7,10]
thanos(A, 8)
```

the length of `A` is 2. (Maybe it would have been better to write `A.thanos(8)`, but Thanos doesn't really worry too much about proper naming conventions for object-oriented programming.)

A search algorithm should present itself: on input `A` and `k`, call `thanos(A, k)` repeatedly until `A` has length `1`, then inspect `A[0]`.

Question 8

Answer saved

Marked out of 1.00

Express the running time $S(n)$ of Thanos-search on inputs of length n as a recurrence relation. Let's say that $S(1) = 1$. (We could elevate the "number of comparisons" to our canonical operation. But it's hard to say what happens inside `thanos` – superhero space magic! –, so it doesn't make much sense to be too precise about this anyway.)

- a. $S(n) = T(n/2) + S(n)$
- b. $S(n) = S(n/2) + 1$
- c. $S(n) = T(n)$
- d. $S(n) = T(n/2) + 1$
- e. $S(n) = T(n) + S(n/2)$
- f. $S(n) = S(n) + T(n)$
- g. $S(n) = S(n/2) + T(n/2)$

Clear my choice

Question 9

Answer saved

Marked out of 1.00

Assume a single call to `thanos(A,k)` takes 1 unit of time – he just snaps his fingers, and it's done. What is the running time of Thanos search?

- a. quadratic in n
- b. logarithmic in n
- c. linearithmic in n
- d. linear in n

Clear my choice

Question 10

Answer saved

Marked out of 1.00

Does `A` have to be sorted for Thanos-search to work?

- a. Thanos' part works either way (literally by magic), but the outer loop needs a linear number of iterations if `A` is not sorted.
- b. Yes if the datatype of `A` is comparable (i.e., implements `Comparable` or `__lt__`). Else no.
- c. No, that's the whole point. Otherwise I could just use binary search and wouldn't need the supervillain–god.
- d. Yes, because every searching data structure implicitly requires its input to be sorted.

Clear my choice