# Binary search Recurrence Relation

① Determine recurrence relation. What is the base case?
— look at the code! Slide # 13

$T(n) = 10 + T(n/2)$ , $T(1) = 10$

$T(n/2) = 10 + T(n/4)$

$T(n/4) = 10 + T(n/8)$

...

② Expand to find general expressi[on] in terms of # expansions: 'K'

exp.

1 $T(n) = 10 + T(n/2)$

2 $= 10 + (10 + T(n/4))$

3 $= 10 + (10 + (10 + T(n/8)))$

...

K $= 10K + T\left(\frac{n}{2^K}\right)$

③ find a value for $k$, such that we hit the base case

Since $T(n) = 1 + T\left(\frac{n}{2^k}\right)$, and $T(1) = 10$

when $\frac{n}{2^k} = 1$, we will enter the base case. So lets solve for $k$.

$$\frac{n}{2^k} = 1$$

$$n = 2^k$$

$$\log(n) = \log_2 2^k$$

$$\log n = k.$$

④ plug value for k in.

$$T(n) = 10k + T\left(\frac{n}{2^k}\right), \quad k = \log n$$

$$= 10\log n + T\left(\frac{n}{2^{\log n}}\right)$$

$$= 10\log n + T\left(\frac{n}{n}\right)$$

$$= 10\log n + \underline{T(1)}$$ base case

$$T(n) = 10\log n + 10$$

so $T(n) = O(\log n)$ (drop low order terms and leading coefficients.

Recursive ArraySum : slide # 19

① Determine recurrence relation

$$T(n) = \underbrace{O(1)}_{\text{constant}} + 2T(n/2) \quad , T(1) = 1$$

$$T(n/2) = O(1) + 2T(n/4)$$

$$T(n/4) = O(1) + 2T(n/8)$$

...

② expand, find general expression in terms of # of expansions.

<u>exp.</u>

1 $\quad T(n) = O(1) + 2T(n/2)$

2 $\quad = O(1) + 2(O(1) + 2T(n/4))$

3 $\quad = O(1) + 2(O(1) + 2(O(1) + 2T(n/8)))$

... $\quad = O(1) + 2O(1) + 4O(1) + 8T(n/8)$

k $\quad = \sum_{i=0}^{k-1} 2^i \cdot O(1) + 2^k T\left(\frac{n}{2^k}\right)$

$$= (2^k - 1) O(1) + 2^k T\left(\frac{n}{2^k}\right)$$

remember, we proved that

$$\sum_{i=0}^{k} 2^i = 2^{k+1} - 1 \ \ldots \ \text{so} \ \sum_{i=0}^{k-1} 2^i = 2^{(k-1)+1} - 1 = 2^k -$$

$$k = O(1)(2^k - 1) + 2^k T\left(\frac{n}{2^k}\right)$$

3) Solve for k.

| | |
|---|---|
| $\frac{n}{2^k} = 1$ | $T(1) = 1$ |
| $n = 2^k$ | |
| $k = \log n$ | |

4) plug k in

$$T(n) = (2^k - 1)O(1) + 2^k T\left(\frac{n}{2^k}\right), \quad k = \log n$$

$$= (2^{\log n} - 1)O(1) + 2^{\log n} T\left(\frac{n}{2^{\log n}}\right)$$

$$= (n-1)O(1) + n\underbrace{T(1)}_{\text{base case}}$$

$$= (n-1)O(1) + n(1)$$

$$= n \cdot O(1) - O(1) + n$$ which is $O(n)$ ü