COP 3502 10/13/25

Wed - more info about Exam
will receive exam back on
~~Wed~~ 10/~~22~~21/2025.
Tues 23
Thurs

TODAY - Recurrence Relations

↳ Goal analyze run-time of recursive algs.

last time: Towers of Hanoi

(1) Binary Search

(2) Several examples of different recurrence relations not necessarily tied to algs.

```
binsearch(arr, low, high)
 if (low > high) return 0;
 int mid = (low+high)/2;
 [ if 3 branches
     binsearch(arr, low, mid-1)
     binsearch(arr, mid+1, high) ]
```

Let n be size of input subarray.

Let $T(n)$ be runtime of binary search on a subarray of n items.

$$T(n) \le \underset{\substack{\uparrow \\ \text{left or} \\ \text{right} \\ \text{rec call}}}{\underline{T\left(\frac{n}{2}\right)}} + \underset{\substack{\uparrow \\ \text{if start} \\ \text{mid calc}}}{O(1)}$$

rec call is on subarray of size $\le \frac{n}{2}$

$$T(n) = \boxed{T\left(\frac{n}{2}\right) + O(1)} \qquad T(1) = O(1)$$

$$= T\left(\frac{n}{4}\right) + \underline{O(1)} + \underline{O(1)}$$

$$= \boxed{T\left(\frac{n}{4}\right) + \underline{2}\, O(1)}$$

$$= T\left(\frac{n}{8}\right) + O(1) + 2\,O(1)$$

$$= \boxed{T\left(\frac{n}{8}\right) + 3 \cdot O(1)}$$

$$T\left(\frac{n}{2}\right) = T\left(\frac{n}{4}\right) + O(1)$$

$$T\left(\frac{n}{4}\right) = T\left(\frac{n}{8}\right) + O(1)$$

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

Let $\frac{n}{2^k} = 1 \rightarrow 2^k = n \rightarrow k = \log_2 n$

Plug in $k = \log_2 n$ in the formula

$$T(n) = T(1) + \log_2 n \cdot O(1)$$

$$T(n) = O(1) + (\log_2 n)\,O(1)$$

$$= \boxed{O(\lg n)}$$

Example 3

$$T(n) = \boxed{2T\left(\frac{n}{2}\right) + O(n)}$$

n

$T(1) = O(1)$

$T\left(\frac{n}{2}\right) = 2T\left(\frac{n}{4}\right) + O\left(\frac{n}{2}\right)$

$T\left(\frac{n}{4}\right) = 2T\left(\frac{n}{8}\right) + O\left(\frac{n}{4}\right)$

$$= 2\left[2T\left(\frac{n}{4}\right) + O\left(\frac{n}{2}\right)\right] + O(n)$$

$$= 4T\left(\frac{n}{4}\right) + \underline{O(n)} + \underline{O(n)}$$

$$= \boxed{4T\left(\frac{n}{4}\right) + 2O(n)}$$

$$= 4\left(2T\left(\frac{n}{8}\right) + O\left(\frac{n}{4}\right)\right) + 2O(n)$$

$$= \underline{8T\left(\frac{n}{8}\right) + O(n) + 2O(n)}$$

$$= \boxed{8T\left(\frac{n}{8}\right) + 3O(n)}$$

k iter, $T(n) = \underline{2^k} T\left(\frac{n}{2^k}\right) + kO(n)$

Let $\frac{n}{2^k} = 1 \rightarrow k = \log_2 n$, $2^k = n$

Plug in this k:

$$T(n) = nT(1) + (\log_2 n)O(n)$$

$$= \underline{O(n)} + \boxed{\underline{(\log_2 n)O(n)}}$$

$$= \boxed{O(n\lg n)}$$

$$T(n) = \boxed{2T(\tfrac{n}{2}) + O(1)}$$
$$= 2\left[2T(\tfrac{n}{4}) + O(1)\right] + O(1)$$
$$= 4T(\tfrac{n}{4}) + 2O(1) + O(1)$$
$$= \boxed{4T(\tfrac{n}{4}) + 3O(1)}$$
$$= 4\left[2T(\tfrac{n}{8}) + O(1)\right] + 3O(1)$$
$$= 8T(\tfrac{n}{8}) + 4O(1) + 3O(1)$$
$$= \boxed{8T(\tfrac{n}{8}) + \underline{7} \cdot O(1)}$$

$T(\frac{n}{2}) = 2T(\frac{n}{4}) + O(1)$

$T(\frac{n}{4}) = 2T(\frac{n}{8}) + O(1)$

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

let $\frac{n}{2^k} = 1 \rightarrow 2^k = n$ and $k = \log_2 n$

Plug this in

$$T(n) = n\,T(1) + (n-1)O(1)$$
$$= n\,O(1) + O(n)$$
$$= \boxed{O(n)}$$

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

$$= 2\left[2T\left(\frac{n}{4}\right) + O\left(\left(\frac{n}{2}\right)^2\right)\right] + O(n^2)$$

$$= 4T\left(\frac{n}{4}\right) + \frac{2}{4}O(n^2) + O(n^2)$$

$$= \boxed{4T\left(\frac{n}{4}\right) + O(n^2)\left[1 + \frac{1}{2}\right]}$$

$$= 4\left[2T\left(\frac{n}{8}\right) + O\left(\left(\frac{n}{4}\right)^2\right)\right] + O(n^2)\left[1 + \frac{1}{2}\right]$$

$$= 8T\left(\frac{n}{8}\right) + \frac{1}{4}O(n^2) + O(n^2)\left[1 + \frac{1}{2}\right]$$

$$= \boxed{8T\left(\frac{n}{8}\right) + O(n^2)\left[1 + \frac{1}{2} + \frac{1}{4}\right]}$$

$$T\left(\frac{n}{2}\right) = 2T\left(\frac{n}{4}\right) + O\left(\left(\frac{n}{2}\right)^2\right)$$

$$T\left(\frac{n}{4}\right) = 2T\left(\frac{n}{8}\right) + O\left(\left(\frac{n}{4}\right)^2\right)$$

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

let $\frac{n}{2^k} = 1 \rightarrow 2^k = n$ $k = \log_2 n$, Plug this in

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

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

$$= nT(1) + 2O(n^2)$$

$$= \boxed{O(n^2)}$$

For all recurrence relations of the form

$$T(n) = A\,T\left(\frac{n}{B}\right) + O(n^k)$$

Cases:

if $\log_B A > k$, then $T(n) = O(n^{\log_B A})$

if $\log_B A = k$, then $T(n) = O(n^k \lg n)$

if $\log_B A < k$, then $T(n) = O(n^k)$

Master Theorem

$$T(n) = 4T\left(\frac{n}{2}\right) + O(1) \quad , \quad T(1) = 1$$

$$= 4\left[4T\left(\frac{n}{4}\right) + O(1)\right] + O(1)$$

$$= \underline{16}\, T\left(\frac{n}{4}\right) + O(1)(1+4)$$

$$= 16\left[4 \cdot T\left(\frac{n}{8}\right) + O(1)\right] + O(1)(1+4)$$

$$= \underline{64}\, T\left(\frac{n}{8}\right) + O(1)(1+4+16)$$

$$T\left(\frac{n}{2}\right) = 4T\left(\frac{n}{4}\right) + O(1)$$

$$T\left(\frac{n}{4}\right) = 4T\left(\frac{n}{8}\right) + O(1)$$

k iter $\quad T(n) = 4^k T\left(\frac{n}{2^k}\right) + O(1)\sum_{i=0}^{k-1} 4^i$

let $\frac{n}{2^k} = 1 \rightarrow 2^k = n$, $k = \log_2 n$,

Plug this in:

$$T(n) = (2^2)^k T(1) + O(1) \cdot \frac{(4^k - 1)}{(4-1)}$$

if

$n = 2^k$

$n^2 = (2^k)^2$

$= (2^2)^k$

$= 4^k$

$$= (2^k)^2 T(1) + O(1) \cdot \left(\frac{4^k - 1}{3}\right)$$

$$= n^2 T(1) + O(n^2) \cdot O(1)$$

$$= O(n^2)$$

$A = 4$, $B = 2$, $k = 0$ $\quad T(n) = A\,T\left(\frac{n}{B}\right) + O(n^k)$

$\log_B A = \log_2 4 = 2. > 0$ so $O(n^{\log_2 4})$

$= O(n^2)$