COP 3502 1/23/24

(0) W OH online (next 2 weeks)

(1) Volunteer - OMG (note WC today)

(2) R- Exam (add online review outline typed link from WC)

(3) Recurrence Relations

(4) exam review

$$T(n) = 2T(n-1)+1 \quad (1), \quad T(1)=1$$

Towers of Hanoi Recurrence

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

$$= 2[2T(n-2)+1]+1$$
$$= 4T(n-2)+2+1$$
$$= 4T(n-2)+3 \quad (2)$$
$$= 4(2T(n-3)+1)+3$$
$$= 8T(n-3)+4+3$$
$$= 8T(n-3)+7 \quad (3)$$

After k iterations we have

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

We know $T(1)=1$, so let $n-k=1 \Rightarrow k = n-1$

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

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

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

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

Binary Search

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

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

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

After k iterations we have

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

Since $T(1) = 1$ let $\frac{n}{2^k} = 1 \Rightarrow n = 2^k \quad k = \log_2 n$;

$$T(n) = T\left(\frac{n}{n}\right) + \log_2 n$$

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

$$= 1 + \log_2 n$$

$$= O(\lg n)$$

$T(n) = 2T(\frac{n}{2}) + n$ ①

$T(1) = 1$

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

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

Merge Sort Recurrence

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

$= 4 \times T(\frac{n}{4}) + n + n$

$= 4T(\frac{n}{4}) + 2n$ ②

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

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

$= 8T(\frac{n}{8}) + 3n$ ③

After k iterations we have

$= 2^k T(\frac{n}{2^k}) + kn$

Since $T(1)=1$, let $\frac{n}{2^k} = 1 \Rightarrow 2^k = n$ $k = \log_2 n$

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

$= n + n\log_2 n$

$= O(n \lg n)$

$$T(n) = 3T(n-1) + 1 \quad ① \qquad T(1) = 1$$

$$= 3\left[3T(n-2) + 1\right] + 1$$

$$= 9T(n-2) + (1+3) \quad ②$$

$$= 9\left[3T(n-3) + 1\right] + (1+3)$$

$$= \underline{27}T(n-3) + (1+3+\underline{9}) \quad ③$$

After k iterations we have

$$T(n) = 3^k T(n-k) + \sum_{i=0}^{k-1} 3^i$$

Since $T(1)=1$, Let $n-k=1 \Rightarrow k = n-1$

$$T(n) = 3^{n-1} T(1) + \sum_{i=0}^{n-2} 3^i$$

$$= \frac{2 \times 3^{n-1}}{2} + \frac{3^{n-1} - 1}{3 - 1}$$

$$= \frac{2 \times 3^{n-1} + 3^{n-1} - 1}{2}$$

$$= \frac{3 \times 3^{n-1} - 1}{2}$$

$$= \boxed{\frac{3^n - 1}{2}}$$

$$T(n) = 3T\left(\frac{n}{2}\right) + n^2 \quad (1) \qquad T(1) = 1$$

$$= 3\left(3T\left(\frac{n}{4}\right) + \left(\frac{n}{2}\right)^2\right) + n^2 \qquad T\left(\frac{n}{2}\right) = 3T\left(\frac{n}{4}\right) + \left(\frac{n}{2}\right)^2$$

$$= 9T\left(\frac{n}{4}\right) + \left(\frac{3}{4}n^2 + n^2\right) \quad (2) \qquad T\left(\frac{n}{4}\right) = 3T\left(\frac{n}{8}\right) + \left(\frac{n}{4}\right)^2$$

$$= 9\left(3T\left(\frac{n}{8}\right) + \left(\frac{n}{4}\right)^2\right) + \left(\frac{3}{4}n^2 + n^2\right)$$

$$= 27T\left(\frac{n}{8}\right) + \left(\frac{9}{16}n^2 + \frac{3}{4}n^2 + n^2\right) \quad (3)$$

After k iterations we have:

$$T(n) = 3^k T\left(\frac{n}{2^k}\right) + n^2 \sum_{i=0}^{k-1} \left(\frac{3}{4}\right)^i$$

We know $T(1) = 1$, let $\frac{n}{2^k} = 1 \Rightarrow n = 2^k$, $k = \log_2 n$

$$T(n) \leq 3^{\log_2 n} T(1) + n^2 \cdot \sum_{i=0}^{\infty} \left(\frac{3}{4}\right)^i$$

$$= n^{\log_2 3} + n^2 \cdot \frac{1}{1 - \frac{3}{4}}$$

$$= n^{\log_2 3} + 4n^2$$

$$= O(n^2)$$

$$a^{\log_b c} = c^{\log_b a}$$

$\downarrow$

See if you can prove this!

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

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

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

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

$$= 8T\left(\frac{n}{8}\right) + \left(\frac{1}{4}n^2 + \frac{1}{2}n^2 + n^2\right) \quad (3)$$

After k iterations we have

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

Since $T(1) = 1$ let $\frac{n}{2^k} = 1$ $2^k = n$, $k = \log_2 n$

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

$$= n + n^2 \cdot \frac{1}{1 - \frac{1}{2}}$$

$$= n + 2n^2$$

$$= O(n^2)$$

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

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

$$= 16\,T\left(\frac{n}{4}\right) + (2n + n)$$

$$= 16\,T\left(\frac{n}{4}\right) + 3n \quad ②$$

$$= 16\left[4T\left(\frac{n}{8}\right) + \frac{n}{4}\right] + 3n$$

$$= 64\,T\left(\frac{n}{8}\right) + 4n + 3n$$

$$= 64\,T\left(\frac{n}{8}\right) + 7n \quad ③$$

After k iterations we have

$$T(n) = 4^k\,T\left(\frac{n}{2^k}\right) + (2^k - 1)n$$

Since $T(1) = 1$, let $\frac{n}{2^k} = 1$ $\quad n = 2^k,\ k = \log_2 n$

if $n = 2^k$ $\quad n^2 = (2^k)^2 = (2^2)^k = 4^k$

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

$$= n^2 + n^2 - n$$

$$= O(n^2)$$

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

Case 1 if $B^k < A$, $O(n^{\log_B A})$

Case 2 if $B^k = A$, $O(n^{\log_B A} \lg n)$

$O(n^k \lg n)$

Case 3 if $B^k > A$, $O(n^k)$

$T(n) = 3T\left(\frac{n}{2}\right) + n^2$ $A=3, B=2, k=2$ $B^k=4$, $A=3$

Case 3: $O(n^2)$

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

Case 3: $O(n^2)$

$T(n) = 4T\left(\frac{n}{2}\right) + n$, $A=4, B=2, k=1$ $B^k=2$ $A=4$

Case 1: $O(n^{\log_2 4})$

$= O(n^2)$

$T(n) = 2T\left(\frac{n}{2}\right) + n$, $A=2, B=2, k=1$, $B^k=2$ $A=2$

Case 2: $O(n^1 \lg n)$

1. Problem Solving
   a. binary search
   b. sorted list matching
2. Dyn Mem
   a. array of primitives
   b. array of struct
   c. array of ptr to struct (benefit of this)
   d. 2D array int
   e. 2D array char (array of strings)
      - same ~~ro~~ room for each str
      - just the right amt mem for each str
      - '\0'
   f. nesting examples (smoothie)
3. ~~Tren~~ Order Analysis Math
   a. summations
   b. timing problems
   c. analyzing code segments
   d. new problem analysis
   e. recurrence relations

Add to target

Input: sorted array, length, target

Output: 1 if 2 diff values in the array sum to the target, 0 if ~~Z~~ ~~no~~ no two values exist

[2, 6, 8, 20, 22, 30, 31, 32, 53]   t = 41

low   high

```
int addtotarget(int* arr, int n, int target) {
    int i = 0, j = n-1;
    while (i < j) {
        if ( arr[i]+arr[j] > target) j--;
        else if (arr[i]+arr[j] < target) i++;
        else return 1;
    }
    return 0;
}
```

```c
char** eachshift (char* str) {
    int len = strlen(str);
    char** res = calloc (26, sizeof(char*));
    for (int i = 0; i<26; i++) {
        res[i] = malloc ((len+1) * sizeof(char));
        for (int j = 0; j<len; j++)
            res[i][j] = 'A' + ((str[j] - 'A') + i) % 26;
        res[i][len] = '\0';
    }
    return res;
}
```

$$\sum_{i=n+1}^{3n} (i+5) = \sum_{i=\phi}^{3n} (i+5) - \sum_{i=\phi}^{n} (i+5)$$

$$= \frac{3n(3n+1)}{2} + 15n - \left[\frac{n(n+1)}{2} + 5n\right]$$

$$= \frac{9n^2+3n}{2} - \left[\frac{n^2+n}{2}\right] + 10n$$

$$= \frac{8n^2+2n}{2} + 10n$$

$$= 4n^2 + n + 10n$$

$$= \boxed{4n^2 + 11n}$$

Program $O(n \log n)$

$n = 10^6$ runtime = 10 sec

How long $n = 10^3$ in milliseconds

$$T(n) = c\,n \log n$$

$$T(10^6) = c \cdot 10^6 \log 10^6 = 10 \text{ sec}$$

$$c = \frac{10 \text{ sec}}{10^6 \times 6 \log 10}$$

$$T(10^3) = \frac{10 \text{ sec}}{10^6 \times 6 \log 10} \times 10^3 \log 10^3$$

$$= \frac{10^4 \text{ms} \times 10^3 \times \underline{3} \times \cancel{\log 10}}{10^6 \times \underline{6} \times \cancel{\log 10}}$$

$$= \frac{10}{2} \text{ ms}$$

$$= 5 \text{ ms}$$

$O(rc^2)$

$r=200 \quad c=500 \qquad$ time $=5$ sec

$r=800 \quad c=300 \qquad$ how long?

$T(\overset{r,c}{A}) = k \cdot rc^2$

$T(200, 500) = k \cdot 200 \cdot 500^2 = 5 \Rightarrow k = \dfrac{5\text{sec}}{200 \times 500^2}$

$T(800, 300) = \dfrac{5\text{sec}}{\cancel{200} \times 500^2} \times \overset{4}{\cancel{800}} \times 300^2$

$= \dfrac{20\text{sec} \times 300^2}{500^2}$

$= 20\text{sec} \times \dfrac{3^2}{5^2} \times \dfrac{\cancel{100^2}}{\cancel{100^2}}$

$= \dfrac{180\text{sec}}{25}$

$= \dfrac{36}{5}$ sec

$= \boxed{7.2 \text{sec}}$

Look over sum neither arithmetic nor geometric

$S = 1 \times \frac{1}{2} + 3 \times \frac{1}{4} + 5 \times \frac{1}{8} + \cdots$

$= \sum_{i=1}^{\infty} (2i-1) \times \left(\frac{1}{2}\right)^i$