CS II Lecture 2

Topics:

Divide and conquer:

- mergesort
- median
- FFT

Mergesort

given array of elements on which an order relation "≤" is defined

$$A = (\overbrace{32,\ 21,\ 15,\ 41,\ 73,\ 13}^{n})$$

output sorted sequence

output 13, 15, 21, 32, 41, 73

A $\quad \underbrace{32,\ 21,\ 15}_{n/2} \qquad \underbrace{41,\ 73,\ 13}_{n/2}$

$\cancel{15}, \cancel{21}, 32 \qquad \cancel{13}, 41, 73$

i $\qquad$ j

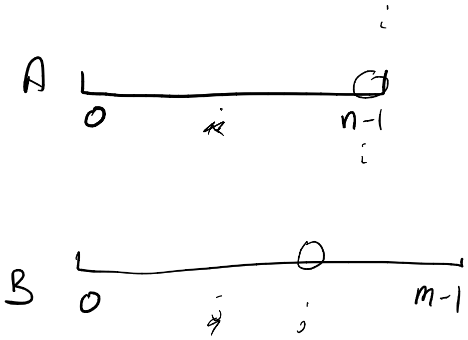
output: 13, 15, 21, 32, 41, 73

k

given
A n elements
B m elements
C initialized array of size n+m

goal: C contain union of A and B, sorted
assuming A sorted, B sorted

```
i = 0
j = 0
k = 0
while i < n and j < m
    if A[i] < B[j]:
        C[k] = A[i]
        k++
        i++
```




```
i = 0
j = 0
k = 0
while (i < n and j < m)
    if A[i] < B[j]:
        C[k] = A[i]
        k++
        i++
    else
        C[k] = B[j]
        k++
        j++
```

O(1) work

n + m iterations

$O(n+m)$ time to merge a sorted sequence of length n with a sorted sequence of length m

Running time

$$T(n) = \underset{a}{2}\, T(n/\underset{b}{2}) + O(n^{\overset{d}{1}})$$

$$T(1) = O(1)$$

$$a = b^d = 2$$

total running time is $O(n \log n)$

Take any algorithm that only accesses data using comparison

Give algorithm a permutation of sequence $1, 2, 3, \ldots, n$

Outcomes of sequence of comparisons done on a permutation π_1 must differ somewhere on the outcome of sequence of comparisons done on π_2

If algorithm makes c comparisons

2^c possible outcomes

$$n! \leq 2^c$$

$$c \geq \log n! \geq n \log n - O(n)$$

32, 12, 17, 41, 30, 12, 17, 32, 25, 42

12, 12, 17, 17, 25, 32, 32, 41, 42

median of sequence of n elements

element that is $\geq$ at least $\lfloor \frac{n}{2} \rfloor$ other elements

and $\leq$ at least $\lfloor \frac{n}{2} \rfloor$ other elements

31 21 15 41 73 13 81

4th element in sorted order

- 31, 21, 15, 41, 13 5

- ~~73, 81~~ 2

reduce to find 4th element in sorted ord of

- 31, 21, 15, 41, 13

- ~~15, 13~~ 2

- 31, 21, 41 3

reduce to find 2-nd element in sorted ord of

- 31, 21, 41

- ~~21~~ 1

- ~~31, 41~~ 2

reduce to find smallest element of

- 31, 41

$$n + n-1 + n-2 + \cdots 2$$

$$O(n^2)$$

select (A, k)

// A array , k integer

// return k-th element of A in sorted order

n= length (A) // A [0,-,n-1]

if n==1

r = random (0,--,n-1)

B = sequence of elements of A that are < A[r]

C = sequence of elements of A that are > A[r]

n_B = length (B)

n_C = length (C)

n_r = # elements of A equal to A[r]

if $0 \le k \le n_B - 1$: return select (B,k)

else if $n_B \le k \le n_r - 1$: return A[r]

else: return select (C, $k - n_B - n_r + 1$)

sorted A =

(sorted B , $\underbrace{A[r] --- A[r]}_{n_r}$, sorted C)

k

$0 \le k \le n_B - 1$

$n_B \le k \le n_B + n_r - 1$

$n_B + n_r \le k \le n - 1$

sorted C

$k - n_B - n_r + 1$

given an input of length n
size of recursive call is a r.v., call it n_1

$T(n)$ running time on input of length n

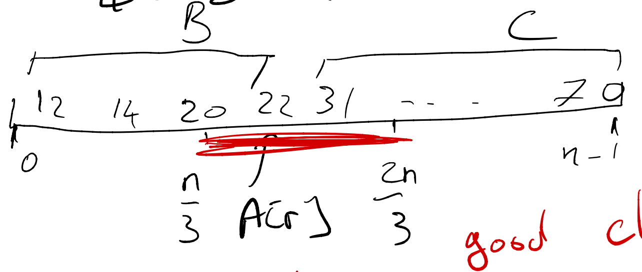
$$T(n) = T(n_1) + O(n)$$

---

call k number of recursive steps that it takes to go from size n to size $\leq \frac{2n}{3}$

$$\mathbb{E}[k] \leq O(1)$$

$$\mathbb{E}[k] \leq \mathbb{E}[\text{\# calls made until} \ldots] \leq \frac{1}{1/3} = 3$$



good choice of r

$\geq \frac{1}{3}$

Prob $\geq \frac{1}{3}$ that $k = 1$

$k \leq$ # recursive calls made until one recursive call chooses a good r

$$\mathbb{E}\, T(n) \le \mathbb{E}\, T(2/3\, n) + \mathbb{E}\,[O(n) \cdot \#\text{ of calls to go from size } n \text{ to size} \le 2n/3]$$

$$\le \mathbb{E}\, T(2/3\, n) + O(n)$$

$$t(n) = t(2/3\, n) + n$$
$$t(1) = 1$$

master theorem

solution is $O(n)$

$$\mathbb{E}\, T(n) = O(n)$$

Fast Fourier Transform

Discrete Fourier Transform

$$f : \{0, 1, 2, 3, \ldots, n-1\} \to \mathbb{C}$$

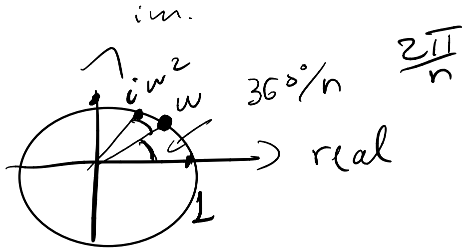
$$w := e^{2\pi i/n}$$

$w^n = 1 = w^0$

$w^0, w^1, w^2, w^3, \ldots, w^{n-1}$ different values

-1 primitive 2nd root of 1

i primitive 4th root of 1



$w^2 = e^{2\pi i \cdot \frac{2}{n}}$

$= \left(e^{2\pi i/n}\right)^2$

for each s in $\{0, \ldots, n-1\}$

$g_s(x) = e^{2\pi i\, s \cdot x/n} = w^{s \cdot x}$ defined for $x = 0, \ldots, n-1$

prove $g_s(\cdot)$ basis for space of functions $f : \{0, \ldots, n-1\} \to \mathbb{C}$

Any function

$$f : \{0, \dots, n-1\} \to \mathbb{C}$$

can be written as linear combination

$$f(x) = \sum_s \hat{f}(s)\, g_s(x)$$

$$= \frac{1}{\sqrt{n}} \sum_{s=0}^{n-1} \hat{f}(s)\, e^{2\pi i \cdot s \cdot x / n}$$

$$\sum_x |f|^2(x) = \sum_s |\hat{f}|^2(s)$$ Parseval's identity

$$\sum_x \overline{g(x)}\, f(x) = \sum_s \overline{\hat{g}(x)}\, \hat{f}(x)$$