ILLINOIS INSTITUTE OF TECHNOLOGY

Transforming Lives. Inventing the Future. www.iit.edu

CS430

Introduction to Algorithms

Lec 7

Lan Yao

Outlines

- **Quick Sort**

Quicksort

- Procedure:
  - Pick one element as the pivot from the array (in our case, it is A[r], the last element of the array);
  - Partition the array into two sub-arrays, while all elements in left sub-array is less than or equal to A[r] and all elements in right sub-array is greater than or equal to A[r].
  - In either sub- array, recursively partitions it.

Quicksort

- Complexity analysis
  - The worst case for partition: $T(n)=T(n-1)+T(0)+\Theta(n)=T(n-2)+\Theta(n-1)=\Theta(n^2)$;

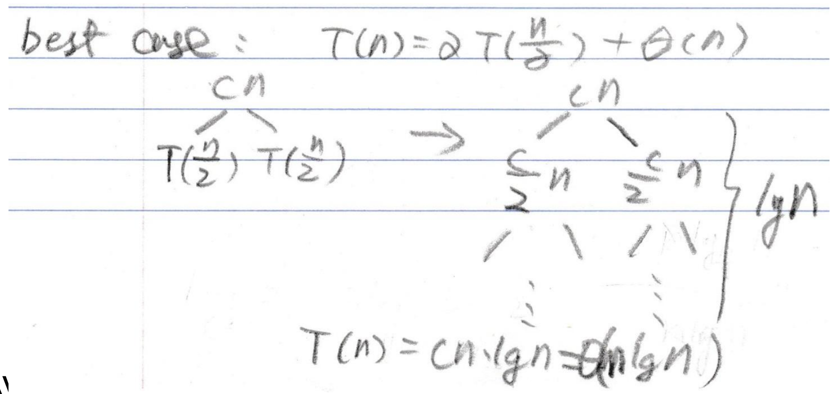
worst cost: $T(n) = T(n-1) + T(0) + \Theta(n)$

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

$$\begin{array}{c} cn \\ | \\ T(n-1) \end{array} \Rightarrow \left.\begin{array}{c} cn \\ | \\ c(n-1) \\ | \\ T(n-2) \\ \vdots \end{array}\right\} n \qquad T(n) = c \cdot \frac{n(n+1)}{2} = \Theta(n^2)$$

Quicksort

- Complexity analysis
  - The best case for partition: T(n)=2T(n/2)+Θ(n)=Θ(nlgn)



  - Av
    - An example: 9-to-1

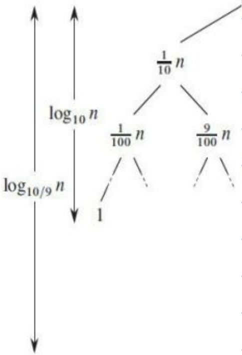
Quicksort

9-to-1 Split tree:

- We do not know the size of each sub-array if they are partitioned by the value of a random element in the array. Let suppose that in each recursion of partition, the array is split at the ratio of 1:9, then we have this 9-to-1 Split tree
- $T(n)=T(9n/10)+T(n/10)+cn$

Quicksort

9-to-1 Split tree:



The number of levels: (9-1- split)

Max: $n\left(\frac{9}{10}\right)^{k} = 1 \Rightarrow k = \log_{\frac{9}{10}} \frac{1}{n} \Rightarrow \log_{\frac{10}{9}} n$

Min: $n \cdot \left(\frac{1}{10}\right)^{k'} \Rightarrow k' = \log_{\frac{1}{10}} \frac{1}{n} = \log_{10} n$

Total cost: (Max) $T(n) = cn\left(\left(\log_{\frac{10}{9}} n + 1\right)\right)$

$= O(n \lg n)$

$$\lim_{n=0}^{\infty} \frac{n \log_{\frac{10}{9}} n}{n \lg n} = \lim_{n=0}^{\infty} \frac{\log_{\frac{10}{9}} n}{\lg n}$$

$$= \lim_{n=0}^{\infty} \frac{\frac{k}{n}}{\frac{k'}{n}} = \frac{k}{k'} \longrightarrow \text{constant !}$$

$\therefore T(n) = cn \log_{\frac{10}{9}} n + cn = \Theta(n \lg n)$

Quicksort

Algorithm:

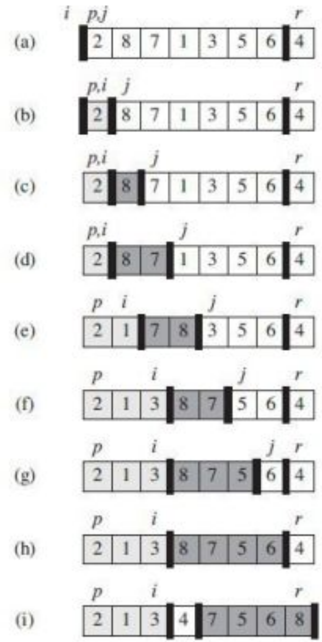
```
QUICKSORT(A, p, r)
if p < r
    q = PARTITION(A, p, r)
    QUICKSORT(A, p, q - 1)
    QUICKSORT(A, q + 1, r)
```

```
PARTITION(A, p, r)
x = A[r]
i = p - 1
for j = p to r - 1
    if A[j] ≤ x
        i = i + 1
        exchange A[i] with A[j]
exchange A[i + 1] with A[r]
return i + 1
```

Quicksort

Algorithm:

```
PARTITION(A, p, r)
x = A[r]
i = p − 1
for j = p to r − 1
    if A[j] ≤ x
        i = i + 1
        exchange A[i] with A[j]
exchange A[i + 1] with A[r]
return i + 1
```



Randomized Version

To allay using A[r] as the pivot, we randomly select an element from the sub-array A[p...r].

```
RANDOMIZED-PARTITION(A, p, r)
i = RANDOM(p, r)
exchange A[r] with A[i]
return PARTITION(A, p, r)
```

Example

Use substitution to show that the running time of Quicksort is $O(n^2)$

proof: $$T(n) = \max_{0 \le q \le n-1} [T(q) + T(n-1-q)] + \Theta(n)$$

step1: guess $T(n)=O(n^2)<=cn^2$

step2: induction proof

$T(n)<=\max[cq^2+c(n-1-q)^2]+\Theta(n)$

Induction Proof

For Quicksort $T(n) = O(n^2) = \max_{0 \le q \le n-1} [T(q) + T(n-1-q)] + \Theta(n)$

Proof: ① Guess: $T(n) = O(n^2)$

② Induction proof:

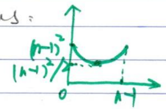
suppose that when n,

$T(n) \le \max_{0 \le q \le n-1} [cq^2 + c(n-1-q)^2] + \Theta(n)$

$\max_{0 \le q \le n-1} [cq^2 + c(n-1-q)^2] = c \max_{0 \le q \le n-1} [\underbrace{2q^2 - 2q(n-1) + (n-1)^2}_{\text{Quadratic}}]$

discriminant $= [-2(n-1)]^2 - 4 \times 2 \times (n-1)^2 < 0$

So plat it as:



The max is $(n-1)^2$.

Plus it in: $T(n) \le c \cdot (n-1)^2 + \Theta(n)$

$= c(n^2 - 2n + 1) + \Theta(n)$

$= cn^2 - c(2n-1) + \Theta(n)$

$= cn^2 - \underbrace{[c(2n-1) - \Theta(n)]}_{\text{positive ?}}$

$\le cn^2$

Therefore: $T(n) \le cn^2$