Quicksort

Wednesday, January 29, 2025 9:51 AM

Last Time:
- Merge Sort
- Recursion Trees

Today:
- Master Theorem
- Quicksort

$\boxed{\text{HW 2}}$!!

$$\boxed{T(n) = 3T(n/4) + \Theta(n)} \longrightarrow \Theta(n)$$

Master Theorem

$T(n) = a \cdot T(n/b) + f(n)$

$a, b \geq 1$, $f(n)$ is any function.

Then:

1. If $f(n)$ is $O(n^{\log_b a - \varepsilon})$ for $\varepsilon > 0$,
   then $T(n)$ is $\Theta(n^{\log_b a})$

2. If $f(n)$ is $\Theta(n^{\log_b a})$
   then $T(n)$ is $\Theta(n^{\log_b a} \log n)$

3. If $f(n)$ is $\Omega(n^{\log_b a + \varepsilon})$, $\varepsilon > 0$
   and $a f(\frac{n}{b}) \leq c \cdot n$ for some c.
   then $T(n)$ is $\Theta(f(n))$

1. $T(n) = 3T(n/4) + \Theta(n)$

$a = 3$
$b = 4$ $\Rightarrow n^{\log_4 3}$ $\qquad f(n) = n^1$

$\boxed{T(n) \text{ is } \Theta(n) \text{ by case 3 of master theorem}}$

master theorem

2. $T(n) = 16\,T(n/4) + \Theta(n^2)$

$a = 16$
$b = 4$ $\Rightarrow n^{\log_4 16} \to n^2$ $\qquad n^2$

$\boxed{T(n) \text{ is } \Theta(n^2 \log n)}$

$T(n) = 2\,T(n-1) + 1$ do recursion tree!!

$T(n) = T(n/3) + T(2n/3) + n^2$

Proof of the Master Theorem?

- p. 97 – 106 in CLRS
- recursion trees

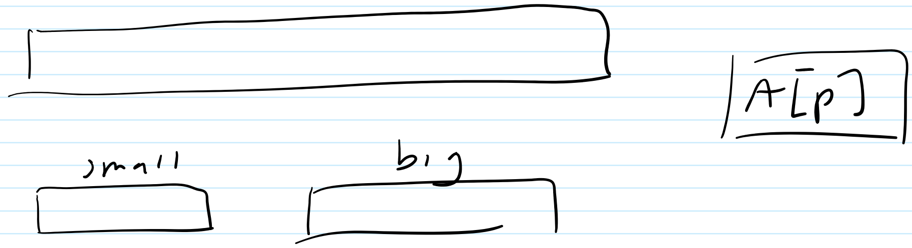
Quick sort $A[1..n]$ Put it in order

Subroutine
partition.
Given a list $A[1...n]$ and a target p.
split the array into 3 parts.

$\boxed{< A[p]}$, $A[p]$, $\boxed{> A[p]}$

$[\ < A[p]\ ],\ A[p],\ [\ > A[p]\ ]$

smaller — larger.




```
Qsort (A[1...n])
    if n ≤ 1 : sorted!
    otherwise:
        choose a pivot index p.
        r ← partition (A[1..n], p)          → Θ(n)
        Qsort (A[1... r-1])
        Qsort (A[r+1 ... n])
```

```
Partition(A[1...n], p)
    swap A[p] with A[n]          } → O(1)
    l ← 0
    for i in 1...(n-1):
        if A[i] < A[n]:          → something to count.
            l ← l+1
            swap A[i] with A[l]
    swap A[l+1] with A[n]        } → O(1)
    return (l+1)
```

Efficiency

$$\boxed{T(n) = \Theta(n) + T(r-1) + T(n-r)}$$

$$\boxed{T(n) = \underset{\substack{\uparrow \\ \text{partition}}}{\theta(n)} + \underline{T(r-1)} + \underline{T(n-r)}}$$

$r \to$ "rank" of the pivot

① Very bad at choosing pivots ...

$r = 1$

$$T(n) = \theta(n) + T(n-1)$$

$$\left.\begin{array}{c} n \\ \downarrow \\ n-1 \\ \downarrow \\ n-2 \\ \downarrow \\ \vdots \end{array}\right\} \to \boxed{O(n^2)} \to \boxed{\text{worst case running time}}$$

How to choose pivots:

1. choose the last item
2. choose the mid item
3. choose the median of 1^{st}, mid, last items.

Ideal pivot? $r = n/2 \to$ $\to$ median

$$T(n) = \theta(n) + T(n/2) + T(n/2)$$

$$T(n) = \theta(n) + 2T(n/2) \implies \underline{\theta(n \log n)}$$

median finding in $\theta(n)$ is possible $\to$ worst case.

Randomized Quicksort

(1) Randomly choose an index p to be

① Randomly choose an index p to be the pivot.

Expected running time.

$$E[T(n)] \longrightarrow \Theta(n \log n)$$

Toss 50 coins!

Expect number of heads?

25

Observations

① partition does most the work.

partition is called at most n times.

② runtime of partition

some parts take O(1) time.

contribute O(n) overall.

if we count the # of comparisons made by partition, this counts the rest of partition's work.

X

X

Lemma Let X be the number of comparisons by partition over the course of quicksort.

Then, $T(n) = O(n + X)$ time.

We will bound X → the number of comparisons.

$E[X]$ → expected

For analysis Let $A = \{z_1, z_2, z_3 \dots z_i \dots z_j \dots z_n\}$

z_i is rank i.

Let Z_{ij} be the set $\{z_i \dots z_j\}$ inclusive.

---

z_i v.s. z_j