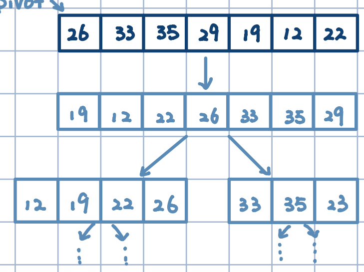
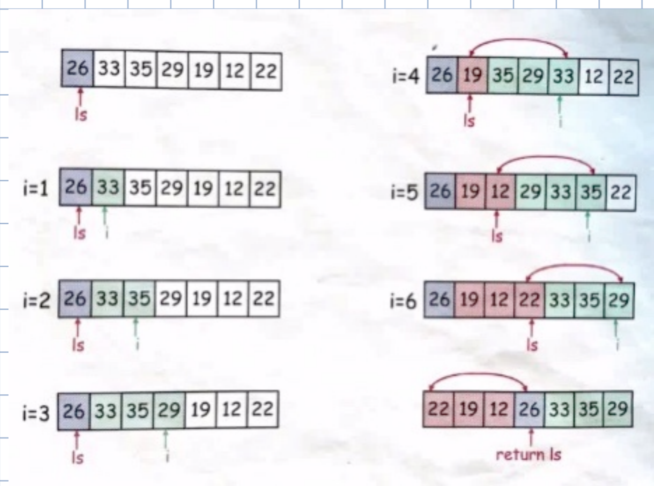
Quicksort

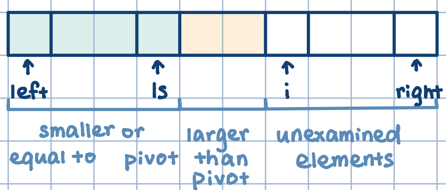
A "divide & conquer" algorithm

```
Quicksort(A, left, right)
    pivot = Partition(A, left, right)
    if (pivot > left)
        Quicksort(A, left, pivot)
    if (pivot + 1 < right)
        Quicksort(A, pivot + 1, right)
```



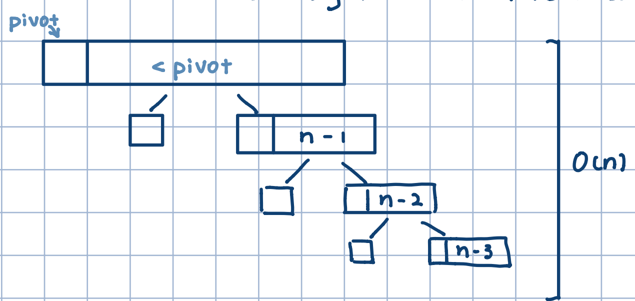

```
int Partition(A, left, right)
    ls = left
    pivot = A(left)
    for i = left + 1 ··· right
        if (A(i) ≤ pivot)
            ls = ls + 1
            swap(A(i), A(ls))
    swap(A(left), A(ls))
    return ls
```





Worst Case Analysis

Worst case is when the array is sorted (or reversed sorted)



$\Rightarrow T(n) = T(n-1) + \Theta(n)$ → Partition

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

$= T(n-3) + \Theta(n-2) + \Theta(n-1) + \Theta(n)$

$= \Theta(\sum_{k=1}^{n} k) = \Theta(n^2)$ ⇒ is quaratic

Best case analysis

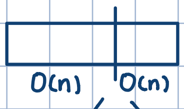
Half element in each partition

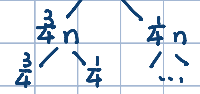
$T(n) = 2T(\frac{n}{2}) + \Theta(n)$

= identical to Mergesort

$= \Theta(n\log n)$

Expected or "Balanced" Partition



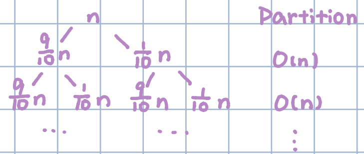


Both partitions are propotional to $O(n)$

Example:

9/10 th one partition and 1/10 th the other

$T(n) = T(\frac{9n}{10}) + T(\frac{n}{10}) + \Theta(n)$ → time to partition



total amount of work $= h\Theta(n)$ → work at every level

$(\frac{9}{10})^h \cdot n = 1 \Leftrightarrow (\frac{10}{9})^h = n \Leftrightarrow h = \log_{\frac{10}{9}}^{n} = O(\log^n)$

$\Rightarrow$ total amount of work $= \Theta(n\log n)$ → but smaller constant

if $T(n) = T(\frac{9999}{10000}n) + T(\frac{1}{10000}n) + O(n) = \Theta(n\log^n)$ } compare

Worst case of deterministic Quicksort (Official)

$T(n) = \max\{T(q) + T(n-q)\} + \Theta(n) \qquad 1 \le q \le n-1$

Use substitution

We guess $T(n) \le cn^2$, substitude to above:

$T(n) \le \max\limits_{1 \le q \le n-1} \{cq^2 + c(n-q)^2\} + \Theta(n)$

$= c\max\limits_{1 \le q \le n-1} \{q^2 + (n-q)^2\} + \Theta(n)$

Achieves at endpoints 1 & n-1 because second derivative with q is positive.

Set $q = 1$, then:

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

For large value of c s.t. it dominates the constant in $\Theta(n)$ and $-2c(n-1) + \Theta(n)$ becomes negative.

Therefore, (officially) $T(n) = \Theta(n^2)$ for worst case

Randomized Quicksort

Randomized_Partition (A, left, right)

i = random (left, right)

swap (A[left], A[i])

return Partition(A, left, right)

Example:

14 10 12 1 15 17 20 → 1 ... 14

Average case for randomized Quick Sort $O(n\log^n)$

$T(n) = \frac{1}{n}(\sum_{q=1}^{n-1} T(q) + T(n-q)) + \Theta(n)$ → Partition

where $T(1) + T(n-1) \times \frac{1}{n} = \Theta(n)$ → absorbed

We know: $\frac{1}{n}(T(1) + T(n+1)) \le \frac{1}{n}(\Theta(1) + \Theta(n))$

$= \Theta(n)$

Therefore, $T(n) = \frac{1}{n}\sum_{q=1}^{n-1}(T(q) + T(n-q)) + \Theta(n)$

$= \frac{2}{n}\sum_{k=1}^{n-1} T(k) + \Theta(n)$ ——— ②

Use substitution (Induction) to show that ② is $an\log^n + b$ for large a & b

Lemma: $\sum_{k=1}^{n-1} k\log^k \le \frac{1}{2}n^2\log^n - \frac{1}{8}n^2$

② $\le \frac{2}{n}\sum_{k=1}^{n-1} ak\log k + b + \Theta(n)$

$= \frac{2a}{n}\sum_{k=1}^{n-1} k\log k + \frac{2b}{n}(n-1) + \Theta(n)$

Use Lemma:

$$\le an\log n - \frac{a}{4}n + 2b + \theta(n)$$

$$= an\log n + b + \underline{\left(\theta(n) + b - \frac{a}{4}n\right)}$$

$$\le an\log n + b$$ For large enough a to make this negative

Proof of Lemma:

$$\sum_{k=1}^{n-1} k\log k = \sum_{k=1}^{\frac{n}{2}-1} k\log k + \sum_{k=\frac{n}{2}}^{n-1} k\log k \le \left(\log\frac{n}{2}\right)\sum_{k=1}^{\frac{n}{2}-1} k + (\log n)\sum_{k=\frac{n}{2}}^{n-1} k$$

$$= (\log n - 1)\sum_{k=1}^{\frac{n}{2}-1} k + (\log n)\sum_{k=\frac{n}{2}}^{n-1} k$$

$$= \log n \sum_{k=1}^{\frac{n}{2}-1} k - \sum_{k=1}^{\frac{n}{2}-1} k + \log n \sum_{k=\frac{n}{2}}^{n-1} k$$

$$= \log n \sum_{k=1}^{n} k - \sum_{k=1}^{\frac{n}{2}-1} k$$

$$\le \frac{1}{2}n(n-1)\log n - \frac{1}{2}\left(\frac{n}{2}-1\right)\frac{n}{2}$$

$$\le \frac{1}{2}n^2\log n - \frac{1}{8}n^2$$