4. Tutorial 3

Monday, February 22, 2021   10:22 AM

Tutorial # 3

① multi-set S of n   algorithm to find k^{th} smallest element
index k (1-n)

→ Quick sort
          pivot → fixed → {sorted left & right}
          assume (k=6)
    L _ _ _ |P| _ _ _ H
             p=5 (6>5)
    ⇒ QS(p+1, H)

* Quick solution algorithm → max (logn) times
              for every logn times, it runs N-times
    N^2 is worst case → O(N)

if one elements position is fixed, N/4 elements are fixed
    → quick

② In an array, 'n elements
determine if a ~~number~~ element occurs more than N/2 times

Quick sort

③ n records, In A
sort in O(n) and no additional space
                                    0 1 0 1 1 0 0 1
① all keys 0/1 → (Quick sort) if 0 → 0000 0 1 1 1 1 ↓
② all keys [0... k]   if 1 → 0000 0 1 1 0 1 1
        ↳ constant   ↗ O(N)
    ↳ 0 1 2 0 2 1 0 0 1 2
              ↳ max k iterations → worst case → O(k.N) ⓜMAXI

④ Sort first 2/3 of A
Sort last 2/3 of A
Sort first 2/3 of A

$$T(N) = 3\left(T\left(\frac{2N}{3}\right)\right) = 9\left(\frac{4N}{9}\right) = 27\left(\frac{8N}{27}\right)$$

$$T\left(\frac{2N}{3}\right) = 3 \cdot \left(T\left(\frac{2}{3} \cdot \frac{2}{3}N\right)\right) = 3\left(\frac{4N}{9}\right)$$

$$T(N) = 3^k \cdot \left(\frac{2^k N}{3^k}\right) \longrightarrow \text{Solved only by using masters' theore}$$
$$\hookrightarrow \text{when } T(1) \text{ is unknown}$$

$$T(N) = aT\left(\frac{N}{b}\right) + f(n)$$
$$f(n) \in n^c$$

1) $c < \log_b a \Rightarrow \Theta\left(n^{\log_b a}\right)$

2) $c = \log_b a \Rightarrow \Theta\left(n^c \cdot n^{\log_k n}\right)$

3) $c > \log_b a \Rightarrow O(f(n))$