Quick Sorting (13)

– Quick Sorting 알고리즘

- Best-Case Analysis
  - best case 는 merge sorting 에서와 같이 나누어지는 부분배열의 크기가 각각 n/2 인 경우이다. 즉, 똑 같은 크기의 두 부분배열로 계속 나누어지는 경우이다.
  - 이 경우의 time complexity 는 merge sorting의 time complexity와 같으므로 그 order는 O($n$log$n$) 이다.

- Average-Case Analysis
  - 배열에 속하는 데이터가 모두 같은 확률로 Pivot 으로 선택된다고 가정하여 분석함.
  - 이 경우의 order는 O($n$log$n$) 임.
  - Average-Case는 worst-case 보다는 best-case 에 가까움
    - » worst-cast가 되는 경우는 매번 pivot을 선택할 때마다 최악의 pivot을 선택해야 만하고 이 경우는 확률적으로 매우 희박함

Quick Sorting (14)

– Average-Case Analysis

- 배열에 속하는 데이터가 모두 같은 확률로 Pivot 으로 선택된다고 가정.
  - pivot이 i-번째 숫자인 경우

$$T(n) = \underbrace{T(i-1)}_{\text{Time to sort left subarray}} + \underbrace{T(n-i)}_{\text{Time to sort right subarray}} + \underbrace{(n-1)}_{\text{Time to partition}}$$

  - pivot이 모두 같은 확률로 i-번째 ($1 \le i \le n$) 숫자인 경우

$$T(n) = \begin{cases} 0 & n = 0 \\ 0 & n = 1 \\ \dfrac{1}{n}\displaystyle\sum_{i=1}^{n}\{T(i-1) + T(n-i)\} + (n-1) & n > 1 \end{cases}$$

Quick Sorting (15)

$$T(n) = \frac{1}{n}\sum_{i=1}^{n}\{T(i-1) + T(n-i)\} + (n-1)$$

$$\sum_{i=1}^{n} T(n-i) = \sum_{i=1}^{n} T(i-1)$$ 이므로

$$T(n) = \frac{2}{n}\sum_{i=1}^{n} T(i-1) + (n-1) = \frac{2}{n}\sum_{i=0}^{n-1} T(i) + (n-1)$$

위 식의 양변에 n을 곱하면

$$nT(n) = 2\sum_{i=0}^{n-1} T(i) + n(n-1)$$

그리고 다음 식을 위 식에서 빼면

$$(n-1)T(n-1) = 2\sum_{i=0}^{n-2} T(i) + (n-1)(n-2)$$

$$nT(n) - (n-1)T(n-1) = 2T(n-1) + 2(n-1)$$

$$nT(n) - (n+1)T(n-1) = 2(n-1)$$

위 식을 n(n+1)로 나누면 다음과 같은 점화식이 된다.

$$\frac{T(n)}{n+1} - \frac{T(n-1)}{n} = \frac{2(n-1)}{n(n+1)} \le \frac{2}{n}$$

Quick Sorting (15)

점화식을 푸는 Telescoping 방법에 의하여

$$\frac{T(n)}{n+1} - \frac{T(n-1)}{n} \le \frac{2}{n}$$

$$\frac{T(n-1)}{n} - \frac{T(n-2)}{n-1} \le \frac{2}{n-1}$$

$$\frac{T(n-2)}{n-1} - \frac{T(n-3)}{n-2} \le \frac{2}{n-2}$$

$$\dots$$

$$\frac{T(2)}{3} - \frac{T(1)}{2} \le \frac{2}{2}$$

위 모든 식을 등호의 좌변과 우변을 더하면

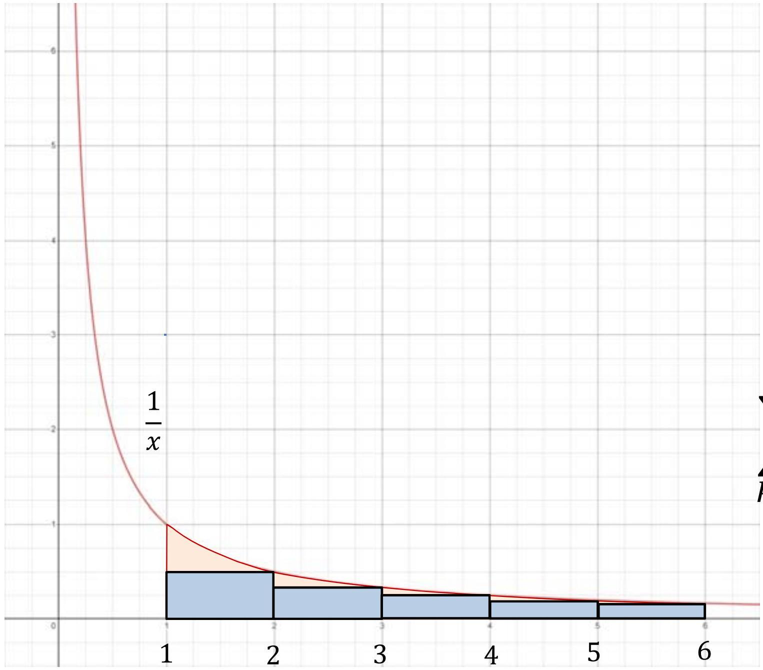
$$\frac{T(n)}{n+1} - \frac{T(0)}{1} \le 2\left(\frac{1}{n} + \frac{1}{n-1} + \dots + \frac{1}{2}\right)$$

$$\frac{T(n)}{n+1} \le 2\sum_{k=2}^{n} \frac{1}{k} \le 2\int_{1}^{n} \frac{1}{x} = 2\ln(n)$$

따라서

$$T(n) \le 2(n+1)\ln(n)$$

$$T(n) = O(n\log n)$$



$$\sum_{k=2}^{n} \frac{1}{k} = \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$$

$$< \int_{1}^{n} \frac{1}{x}$$

$$= \log_{e} n$$

Quick Sorting (16)

- Space Complexity of Quick Sorting
  - Quick Sorting implemented by Recursion
  - How many activation records are stacked at call stack?
    - worst case: O(log n)

pivot에 의해서 나누어진 left part와 right part 중에서 더 짧은 part를 먼저 재귀적으로 실행함

- 더 긴 part를 먼저 실행하게 되면 O(n) space 가 필요함
  - time-complexity 가 worst-case 인 경우
  - 그러나, 짧은 pat를 먼저 재귀적으로 실행하면 O(1) space가 필요함
- time-complexity가 best-case 인 경우에 space가 O(log n) 만큼 필요함

- Quick sorting
  - not an in-place algorithm