CS580 Algorithm Design, Analysis, and Implementation

Lecture notes, Jan 18, 2022
Wufei Ma

1 Quick Sort

Given an array of elements `A`.

1. Divide: split `A[p...r]` into `A[p...q]` and `A[q+1...r]`.
2. Recurse: sort `A[p...q]` and `A[q+1...r]`.
3. Combine: automatic.

```
Procedure QuickSort(p, r):
    if p < r:
        q := Partition(p, r)
        QuickSort(p, q)
        QuickSort(q+1, r)
```

Partition. The objective is to split `A` into a left and a right part so that all numbers in the left part are less or equal to all numbers in the right.

```
Procedure Partition(p, r):
    x := A[p]
    i = p - 1
    j = r + 1
    while True:
        while A[j] > x:
            j = j - 1
        while A[i] < x:
            i = i + 1
        if i < j:
            swap(A[i], A[j])
        else:
            return j
```

- If `A[p]` is the smallest, then the partition is `p` and the recursion continues with two arrays with length 1 and $r - p$ respectively.
- If `A[p]` is K-smallest, for $K \geq 2$, then partition is $K - 1$ and the recursion continues with two arrays with length $k - 1$ and $(r - p + 1) - (k - 1) = r - p - k + 2$ respectively.

Worst case. In the worst case, each partition splits the array of length $n-1$ into two arrays of length 1 and $n-1$. The time complexity is:

$$\begin{aligned} T(n) &= T(n-1) + T(1) + n \\ &= n + (n-1) + \cdots + 2 + n \\ &= \frac{n(n+1)}{2} + (n-1) = \Theta(n^2) \end{aligned}$$

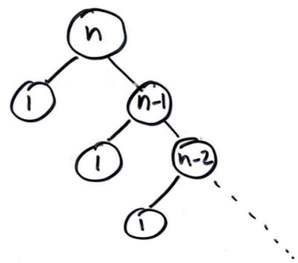

Figure 1: Worst case.

Best case. In the best case, each partition splits the array into two arrays of equal lengths (or differs by 1).

$$\begin{aligned} T(n) &= 2T(n/2) + n \\ &= 2 \cdot (2T(n/4) + n/2) + n \\ &= 4 \cdot T(n/4) + 2n \\ &= \cdots \\ &= n \cdot T(1) + n \log n \\ &= \Theta(n \log n) \end{aligned}$$

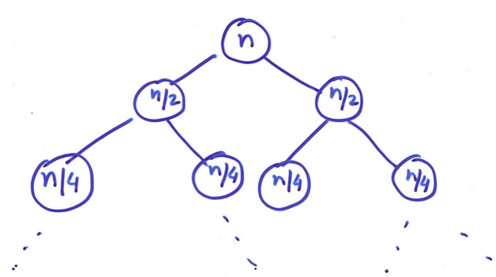

Figure 2: Best case.

Randomized quick sort.

```
Procedure Random-Partition(p, r):
    i := Random(p, r)
    swap(A[p], A[i])
    return Partition(p, r)
```

The complexity is given by the **average analysis**. We have

$$\begin{aligned} T(n) &= \frac{1}{n}\sum_{q=1}^{n-1}\left(T(q)+T(n-q)+\Theta(n)\right) \\ &= \frac{2}{n}\sum_{k=1}^{n-1}T(k)+\Theta(n) \end{aligned}$$

We prove by induction. Assume $T(n) \leq an\log n + b$ for $n = 1, \dots, n-1$. It follows that

$$\begin{aligned} T(n) &\leq \frac{2}{n}\sum_{k=1}^{n-1}(ak\log k + b) + \Theta(n) \\ &= \frac{2a}{n}\sum_{k=1}^{n-1}k\log k + \frac{2b(n-1)}{n} + \Theta(n) \end{aligned}$$

Since

$$\sum_{k=1}^{n-1}k\log k \leq \frac{1}{2}n^2\log n - \frac{n^2}{8}$$

We have

$$\begin{aligned} T(n) &\leq an\log n - \frac{an}{4} + 2b + \Theta(n) \\ &= an\log n + b + (\Theta(n) + b - \frac{an}{4}) \\ &\leq an\log n + b \end{aligned}$$

if we have

$$\begin{aligned} \Theta(n) + b - \frac{an}{4} &\leq 0 \\ a &\geq 4c + 4b \end{aligned}$$

Note that c is given from $\Theta(n)$, we can guarantee this inequality by properly choosing the values of a and b.