Part6-QuickSort-Analysis

Completed on	@2020/03/20
Key videos	https://www.coursera.org/learn/algorithms-divide-conquer/lecture/slY1s/analysis-ii-the-key-insight https://www.coursera.org/learn/algorithms-divide-conquer/lecture/aqD1O/analysis-iii-final-calculations
Note	In-depth analysis of QuickSort algorithm from the perspective of the Decomposition Principle and linearity of expectation.

Note

▼ Analysis I: A Decomposition Principle

- Necessary background

  Assumption: you know and remember (finite) sample spaces, random variables, expectation, linearity of expectation

- Average running time

  从概率的角度去考虑快排的效率

  <u>QuickSort Theorem</u> : for every input array of length n, the average running time of QuickSort (with random pivots) is O(nlog(n)).

  <u>Note</u> : holds for every input. [no assumptions on the data]
  - recall our guiding principles !
  - "average" is over random choices made by the algorithm (i.e., the pivot choices)

- Preliminaries：对于数组n，样本空间Omega，快排算法整体占主导的复杂度来源于算法比较

  ⇒ 因此从数学期望的角度计算平均算法复杂度为O(nlogn)

Fix input array A of length n

Sample Space Ω = all possible outcomes of random choices in QuickSort (i.e., pivot sequences)

Key Random Variable : for $\sigma \in \Omega$

$C(\sigma)$ = # of comparisons between two input elements made by QuickSort (given random choices σ)

Lemma: running time of QuickSort dominated by comparisons.

Remaining goal : E[C] = O(nlog(n))

There exist constant c s.t. for all $\sigma \in \Omega$, $RT(\sigma) \leq c \cdot C(\sigma)$ (see notes)

Tim Roughgarden

- Building blocks:

  不能直接应用master method，由于子问题的数量与大小不确定！

  Note can’t apply Master Method [random, unbalanced subproblems]

  [A = final input array]

6	8	10	2
z_2	z_3	z_4	z_1

  Notation : z_i = i^{th} smallest element of A

  For $\sigma \in \Omega$, indices i< j

  $X_{ij}(\sigma)$ = # of times z_i,z_j get compared in QuickSort with pivot sequence σ

  任意两个数比较的次数只有1次（如果其中一个为pivot）或是0次

  Fix two elements of the input array. How many times can these two elements get compared with each other during the execution of QuickSort?

  - 1
  - 0 or 1
  - 0, 1, or 2
  - Any integer between 0 and $n-1$

  Reason : two elements compared only when one is the pivot, which is excluded from future recursive calls.

  Thus : each X_{ij} is an “indicator” (i.e., 0-1) random variable

- A Decomposition Principle

因此总的比较次数的数学期望，只有在于与pivot比较的次数

A Decomposition Approach

So : $C(\sigma)$ = # of comparisons between input elements

$X_{ij}(\sigma)$ = # of comparisons between z_i and z_j

Thus : $\forall \sigma, C(\sigma) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij}(\sigma)$

complicated simple

By Linearity of Expectation : $E[C] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} E[X_{ij}]$

Since $E[X_{ij}] = 0 \cdot Pr[X_{ij} = 0] + 1 \cdot Pr[X_{ij} = 1] = Pr[X_{ij} = 1]$

Thus : $E[C] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} Pr[z_i, z_j \ get \ compared] \quad (*)$

Next video

- 一般方法

A General Decomposition Principle

1. Identify random variable Y that you really care about

2. Express Y as sum of indicator random variables :

$$Y = \sum_{l=1}^{m} X_e$$

3. Apply Linearity of expectation :

$$E[Y] = \sum_{l=1}^{m} Pr[X_e = 1]$$

"just" need to understand these!

▼ Analysis II: The Key Insight

- Key claim

$C(\sigma)$ = # of comparisons between input elements
$X_{ij}(\sigma)$ = # of comparisons between z_i and z_j

i^{th}, j^{th} smallest entries in array

Recall : $E[C] = \sum_{i=1}^{n-1} \sum_{k=i+1}^{n} Pr[X_{ij} = 1]$ $= Pr[z_i \ z_j \ get \ compared]$

Key Claim : for all i < j, Pr[z_i, z_j get compared] = 2/(j-i+1)

$$P(z_i, z_j \text{get compared}) = \frac{2}{j-i+1}$$

- Proof of key claim

分为两种情况：z_i与z_j被选为pivot，则1次比较；反之，则0次比较

Proof of Key Claim

Pr[z_i,z_j get compared] = 2/(j-i+1)

Fix z_i, z_j with i < j
Consider the set $z_i, z_{i+1}, \ldots, z_{j-1}, z_j$

Inductively : as long as none of these are chosen as a pivot, all are passed to the same recursive call.

Consider the first among $z_i, z_{i+1}, .., z_{j-1}, z_j$ that gets chosen as a pivot.

1. If z_i or z_j gets chosen first, then z_i and z_j get compared
2. If one of $z_{i+1}, \ldots, z_{j-1}$ gets chosen first then z_i and z_j are never compared [split into different recursive calls]

KEY INSIGHT

- Proof of key claim (con'd)

1. z_i or z_j gets chosen first => they get compared
2. one of $z_{i+1},\ldots,z_{j-1}$ gets chosen first => z_i , z_j never compared

Note : Since pivots always chosen uniformly at random, each of $z_i,z_{i+1},\ldots,z_{j-1},z_j$ is equally likely to be the first

$\Rightarrow$ Pr[z_i,z_j get compared] = 2/(j-i+1)

Choices that lead to case (1)

Total # of choices

So : $E[C] = \sum_{i=1}^{n-1}\sum_{j=1}^{n}\frac{2}{j-i+1}$

[Still need to show this is O(nlog(n))

结合一二节，得到期望总公式！

▼ Analysis III: Final Calculations

- The story so far：把求和算式展开，然后右端简化放大

$$E[C] = 2\sum_{i=1}^{n-1}\sum_{j=1}^{n}\frac{1}{j-i+1}$$

How big can this be ?

(*)

<= n choices for i

$\theta(n^2)$ $terms$

Note : for each fixed i, the inner sum is

$$\sum_{j=i+1}^{n}\frac{1}{j-i+1} = 1/2 + 1/3 + \ldots$$

$$So \quad E[C] \le 2 \cdot n \cdot \sum_{k=2}^{n}\frac{1}{k}$$

Claim : this is <= ln(n)

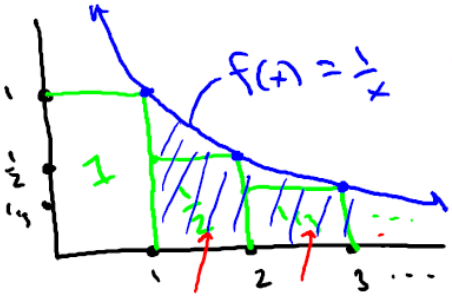
- Completing the proof

将求和转化为积分，最终可得到lnN

$$E[C] \le 2 \cdot n \cdot \sum_{k=2}^{n} \frac{1}{k}$$

$$Claim \quad \sum_{k=2}^{n} \frac{1}{k} \le \ln n$$

<u>Proof of Claim</u>



$$So \quad \sum_{k=2}^{n} \frac{1}{n} \le \int_{1}^{n} \frac{1}{x} dx$$

$$= \ln x \,\Big|_1^n$$

$$= \ln n - \ln 1$$

$$= \ln n$$

Q.E.D. (CLAIM

<u>So</u> :
E[C] <=
2n ln n

Q.E.D.

References

- 分解 (计算机科学)

[Decomposition (computer science)](https://en.wikipedia.org/wiki/Decomposition_(computer_science))
https://en.wikipedia.org/wiki/Decomposition_(computer_science)

[分解 (計算機科學)](https://zh.wikipedia.org/wiki/%E5%88%86%E8%A7%A3_(%E8%A8%88%E7%AE%97%E6%A9%9F%E7%A7%91%E5%AD%B8))
https://zh.wikipedia.org/wiki/%E5%88%86%E8%A7%A3_(%E8%A8%88%E7%AE%97%E6%A9%9F%E7%A7%91%E5%AD%B8)