Revisiting Calculation of Moments of Number of Comparisons used by the Randomized Quick Sort Algorithm

by

Sumit Kumar Jha

in

Discrete Mathematics, Algorithms and Applications

Report No: IIIT/TR/2016/-1

Centre for Security, Theory and Algorithms
International Institute of Information Technology
Hyderabad - 500 032, INDIA
October 2016

Revisiting Calculation of Moments of Number of Comparisons used by the Randomized Quick Sort Algorithm

Sumit Kumar Jha
Email: kumarjha.sumit@research.iiit.ac.in
Center for Security, Theory, and Algorithmic Research
International Institute of Information Technology, Hyderabad, India
Preprint of a Paper Accepted
in Discrete Mathematics, Algorithms and Applications
Link: http://www.worldscientific.com/doi/abs/10.1142/S179383091750001X

Abstract

We revisit the method of Kirschenhofer, Prodinger and Tichy to calculate the moments of comparisons used by the randomized quick sort algorithm. We reemphasize that this approach helps in calculating these quantities with less computation. We also point out that as observed by Knuth this method also gives moments for total path length of a binary search tree built over a random set of n keys.

1 Introduction

Consider the following variant of quick sort algorithm from [SW11]: the quick sort algorithm recursively sorts numbers in an array by partitioning it into two smaller and independent subarrays, and thereafter sorting these parts. The partitioning procedure chooses the last element in the array as *pivot* and puts it in its right place where numbers to the left of it are smaller than it, and those to its right are larger than it.

For purposes of this analysis assume that the input array to the quick sort algorithm contains distinct numbers which are randomly ordered. We may assume the input to the algorithm is simply a permutation of $\{1, 2, \cdots, n\}$ (if the input array has n elements).

Let S_n be the set of all $n!$ permutations of $\{1, 2, \cdots, n\}$. Consider a uniform probability distribution on the set S_n, and define for all $\sigma \in S$, $C_n(\sigma)$ to be the number of comparisons used to sort σ by the quick sort algorithm. We wish to calculate mean and variance of C_n over the uniform distribution on S_n.

Our aim here is to obtain following [KHT87] [KHT]

Theorem 1.1 (Knuth [Knu98]). *We have*

$$\text{Mean}(C_n) = 2((n + 1)H_n - n);$$

and

$$\text{Var}(C_n) = 7n^2 - 4(n + 1)^2 H_n^{(2)} - 2(n + 1)H_n + 13n,$$

over the uniform probability distribution on S_n. Here we have used the notation $H_n = \sum_{k=1}^{n} \frac{1}{k}$ and $H_n^{(2)} = \sum_{k=1}^{n} \frac{1}{k^2}$.

Before proceeding we would like to point out that Hennequin [Hen91] has computed the first five cumulants of the number of comparisons of Quicksort.

2 Calculation of mean and variance

Let $a_{n,s}$ be the number of permutations of n elements requiring a total of s comparisons to sort by the procedure of quicksort.

We start by defining the corresponding *probability generating function*:

$$G_n(z) = \sum_{k\geq 0} \frac{a_{n,k}\, z^k}{n!}.$$

Theorem 2.1. *For $n \geq 1$*

$$G_n(z) = \frac{z^{n+1}}{n} \sum_{1\leq j\leq n} G_{n-j}(z)G_{j-1}(z), \tag{1}$$

and

$$G_0(z) = 1. \tag{2}$$

Proof. The first partitioning stage requires $n+1$ comparisons (for some other variants this might be $n-1$). If the pivot element is kth largest, the sub arrays after partitioning are of size $k-1$ and $n-k$. Thus we can write

$$a_{n,s} = \sum_{1\leq k\leq n} \binom{n-1}{k-1} \sum_{i+j=s-(n+1)} a_{n-k,i}\, a_{k-1,j}. \tag{3}$$

Multiplying equation (3) by z^s and dividing by $n!$ we get

$$\frac{a_{ns}\, z^s}{n!} = \sum_{1\leq k\leq n} \frac{z^s}{n} \sum_{i+j=s-(n+1)} \frac{a_{n-k,i}}{(n-k)!} \cdot \frac{a_{k-1,j}}{(k-1)!}$$

$$= \sum_{1\leq k\leq n} \frac{z^s}{n} \cdot \left\{\text{coefficient of } z^{s-(n+1)} \text{ in } G_{n-k}(z)\cdot G_{k-1}(z)\right\}$$

$$= \sum_{1\leq k\leq n} \frac{z^{n+1}}{n} \cdot z^{s-(n+1)} \left\{\text{coefficient of } z^{s-(n+1)} \text{ in } G_{n-k}(z)\cdot G_{k-1}(z)\right\}$$

after which summing on s gives us equation (1). □

We will now consider the *double generating function* $H(z,u)$ defined by

$$H(z,u) = \sum_{n\geq 0} G_n(z)u^n. \tag{4}$$

Corollary 2.2. *We have*

$$\frac{\partial H(z,u)}{\partial u} = z^2 \cdot H^2(z, zu), \tag{5}$$

and

$$H(1,u) = (1-u)^{-1}. \tag{6}$$

Proof. From equation (1) we have

$$\frac{\partial H(z,u)}{\partial u} = z^2 \sum_{n\geq 1} (uz)^{n-1} \sum_{1\leq j\leq n} G_{n-j}(z) G_{j-1}(z)$$

$$= z^2 \sum_{n\geq 1} (uz)^{n-1} \cdot \{\text{coefficient of } (uz)^{n-1} \text{ in } H(z,uz)\cdot H(z,uz)\}$$

$$= z^2 \cdot H(z,zu) \cdot H(z,zu).$$

Equation (6) follows from the fact that $G_n(1) = 1$. □

Now we write the sth factorial moments $\beta_s(n)$ of the random variable with the aid of the probability generating function $G_n(z)$:

$$\beta_s(n) = \left[\frac{d^s}{dz^s} G_n(z)\right]_{z=1}. \tag{7}$$

The generating functions $f_s(u)$ of $\beta_s(n)$ are

$$f_s(u) = \sum_{n\geq 0} \beta_s(n) u^n. \tag{8}$$

By Taylor's formula and equation (7) we get

$$H(z,u) = \sum_{s\geq 0} f_s(u) \frac{(z-1)^s}{s!}. \tag{9}$$

Theorem 2.3. *For integer $s \geq 0$ we have*

$$f_s'(u) = s! \cdot \sum_{i+j+k+l+m=s} \frac{a_i \cdot f_j^{(k)}(u) \cdot f_l^{(m)}(u) \cdot u^{k+m}}{j! \cdot k! \cdot l! \cdot m!}, \tag{10}$$

where

$$a_k = \begin{cases} 1 & \text{if } k = 0; \\ 2 & \text{if } k = 1; \\ 1 & \text{if } k = 2; \\ 0 & \text{if } k > 2. \end{cases}$$

Proof. Using Taylor's theorem we can write

$$f_j(x) = \sum_{k\geq 0} \frac{f_j^{(k)}(u)(x-u)^k}{k!}$$

which on substituting $x = uz$ gives

$$f_j(uz) = \sum_{k\geq 0} \frac{f_j^{(k)}(u)(z-1)^k u^k}{k!}. \tag{11}$$

Now substituting equation (9) in equation (5) gives:

$$\sum_{s\geq 0} f'_s(u)\frac{(z-1)^s}{s!} = z^2 \cdot \sum_{p\geq 0} f_p(uz)\frac{(z-1)^p}{p!} \cdot \sum_{r\geq 0} f_r(uz)\frac{(z-1)^r}{r!}$$

$$= \sum_{i\geq 0} a_i(z-1)^i \cdot \sum_{p\geq 0} \frac{(z-1)^p}{p!} \sum_{l\geq 0} \frac{f_p^{(l)}(u)(z-1)^l u^l}{l!} \cdot \sum_{r\geq 0} \frac{(z-1)^r}{r!} \sum_{m\geq 0} \frac{f_r^{(m)}(u)(z-1)^m u^m}{m!}$$

$$= \sum_{h\geq 0} (z-1)^h \sum_{i+j+k+l+m=h} a_i \cdot \frac{1}{j!} \cdot \frac{f_j^{(k)}(u)\, u^k}{k!} \cdot \frac{1}{l!} \cdot \frac{f_l^{(m)}(u)\, u^m}{m!}$$

where in the second last line we replaced z^2 by $\sum_{i\geq 0} a_i(z-1)^i$. Now comparing coefficients on both sides of the equation gives

$$f'_s(u) = s! \cdot \sum_{i+j+k+l+m=s} \frac{a_i \cdot f_j^{(k)}(u) \cdot f_l^{(m)}(u) \cdot u^{k+m}}{j! \cdot k! \cdot l! \cdot m!}.$$

□

Remark 2.4. *For asymptotic theory of differential equations originating here we recommend reader the paper [CHT02].*

Corollary 2.5. *We have*

$$f_0(u) = (1-u)^{-1}, \tag{12}$$

$$f_1(u) = \frac{2}{(1-u)^2} \log \frac{1}{1-u}, \tag{13}$$

$$f_2(u) = \frac{8\log^2(1-u)}{(1-u)^3} - \frac{8\log(1-u)}{(1-u)^3} - \frac{4\log^2(1-u)}{(1-u)^2} + \frac{12\log(1-u)}{(1-u)^2} + \frac{6}{(1-u)^3} - \frac{6}{(1-u)^2}. \tag{14}$$

Proof. The equation (12) follows from the fact that $\beta_0(n) = 1$ for all $n \geq 0$.

Setting $s = 1$ in equation (10) gives

$$\begin{aligned} f'_1(u) &= a_1 \cdot f_0^{(0)}(u) \cdot f_0^{(0)}(u) + f_0^{(1)}(u) \cdot f_0^{(0)}(u) \cdot u + f_0^{(1)}(u) \cdot f_0^{(0)}(u) \cdot u \\ &\quad + f_1^{(0)}(u) \cdot f_0^{(0)}(u) + f_0^{(0)}(u) \cdot f_1^{(0)}(u) \\ &= \frac{2}{(1-u)^2} + \frac{u}{(1-u)^3} + \frac{u}{(1-u)^3} + \frac{f_1(u)}{(1-u)} + \frac{f_1(u)}{(1-u)}, \end{aligned}$$

where we used the fact that $f_0(u) = (1-u)^{-1}$. The above equation is

$$f'_1(u) - \frac{2f_1(u)}{1-u} = \frac{2u}{(1-u)^3} + \frac{2}{(1-u)^2}. \tag{15}$$

Solving the linear differential equation (15) by multiplying with integrating factor $(1-u)^2$ gives

$$f_1(u) = \frac{2}{(1-u)^2} \log \frac{1}{1-u} + f_1(0) = \frac{2}{(1-u)^2} \log \frac{1}{1-u}.$$

Plugging $s = 2$ in (10) and solving the resultant differential equation gives

$$f_2(u) = \frac{8\log^2(1-u)}{(1-u)^3} - \frac{8\log(1-u)}{(1-u)^3} - \frac{4\log^2(1-u)}{(1-u)^2} + \frac{12\log(1-u)}{(1-u)^2} + \frac{6}{(1-u)^3} - \frac{6}{(1-u)^2}.$$

□

Corollary 2.6. *We have*

$$\beta_1(n) = 2((n+1)H_n - n),$$

and

$$\beta_2(n) = 4(n+1)^2(H_n^2 - H_n^{(2)}) + 4(n+1)^2 H_n - 8(n+1)H_n + 8nH_n - 4nH_n(5+3n) + 11n^2 + 15n.$$

Proof. We use following expansions from [GK07]

$$\frac{1}{(1-u)^{m+1}} \log\left(\frac{1}{1-u}\right) = \sum_{n\geq 0}(H_{n+m} - H_m)\binom{n+m}{n}u^n;$$

$$\frac{1}{(1-u)^{m+1}} \log^2\left(\frac{1}{1-u}\right) = \sum_{n\geq 0}((H_{n+m} - H_m)^2 - (H_{n+m}^{(2)} - H_m^{(2)}))\binom{n+m}{n}u^n,$$

to conclude the assertion. □

Proof of Theorem 1.1. We conclude the results after noting

$$\mathrm{Mean}(C_n) = \beta_1(n),$$

$$\mathrm{Var}(C_n) = \beta_2(n) - (\beta_1(n))^2 + \beta_1(n).$$

□

3 Similar Partial Differential Functional Equations

We point out that following two examples from [KS97] can be analyzed using the method employed here:

1. Moments of total path length L_n of a binary search tree built over a random set of n keys can be extracted from the

$$\frac{\partial L(z,u)}{\partial z} = L^2(zu, u), \quad \frac{\partial L(0,u)}{\partial z} = 1,$$

   where $L(z,u) = \sum_{n\geq 0} L_n(u)z^n$ is the bivariate generating function.

2. A *digital* search tree for which the bivariate generating function $L(z,u)$ satisfies

$$\frac{\partial L(z,u)}{\partial z} = L^2\left(\frac{1}{2}zu, u\right),$$

   with $L(z,0) = 1$.

References

[CHT02] Hua-Huai Chern1, Hsien-Kuei Hwang, and Tsung-Hsi Tsai. An asymptotic theory for cauchy-euler differential equations with applications to the analysis of algorithms. http://www.stat.sinica.edu.tw/chonghi/Papers/An_asymptotic_theory_for_Cauchy-Euler_differential_equations.pdf, 2002.

[GK07] Daniel H. Greene and Donald E. Knuth. *Mathematics for the Analysis of Algorithms: Modern Birkhuser Classics.* Birkhauser Basel, 3rd edition, 2007.

[Hen91] Pascal Hennequin. Analyse en moyenne d'algorithmes, tri rapide et arbres de recherche. http://algo.inria.fr/AofA/Research/src/Hennequin.PhD.html/, 1991.

[Ili13] Vasileios Iliopoulos. The quicksort algorithm and related topics. http://repository.essex.ac.uk/13266/, 2013.

[KHT] P. Kirschenhofer, H. Prodinger, and R.F. Tichy. Ober einige funktionaldifferentialgleichungen aus der analyse von algorithmen. http://math.sun.ac.za/~hproding/pdffiles/hlawka_funktionalgleichungen.pdf.

[KHT87] P. Kirschenhofer, H. Prodinger, and R.F. Tichy. A contribution to the analysis of in situ permutation. *Glasnik Mathematicki*, 22(42):269–278, 1987.

[Knu98] Donald E. Knuth. *The Art of Computer Programming, Volume 3: (2Nd Ed.) Sorting and Searching.* Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 1998.

[KS97] Charles Knessl and Wojciech Szpankowskit. Quicksort algorithm again revisited. http://docs.lib.purdue.edu/cgi/viewcontent.cgi?article=2351&context=cstech, 1997.

[SW11] Robert Sedgewick and Kevin Wayne. *Algorithms.* Addison-Wesley Professional, 4th edition, 2011.