AVERAGE-CASE ANALYSIS OF QUICKSORT

(Lecture notes written by Vašek Chvátal)

1 Introduction

Sorting algorithms. The input is an array of *records*; each record consists of a *key* and a *satellite*; the key is an identifier (typically a number or a string of characters) coming from some linearly ordered set; the satellite is either the information indexed by the key or a pointer to a file that contains the information. The linear order $\prec$ on the universe of all keys is specified by a function `cmp` that, given a pair of pointers $\&R, \&S$ to records R, S, returns a number: with K_X standing for the key in a record X, we have

$$\begin{array}{ll} \texttt{cmp}(\&R,\&S) < 0 & \text{whenever } K_R \prec K_S \ , \\ \texttt{cmp}(\&R,\&S) = 0 & \text{whenever } K_R = K_S \ , \\ \texttt{cmp}(\&R,\&S) > 0 & \text{whenever } K_R \succ K_S \ . \end{array}$$

The job of a sorting algorithm is to permute (if necessary) the records of the input array

$$A[1],\ A[2],\ \ldots,\ A[n]$$

so that the resulting output array $A[1]$, $A[2]$, ..., $A[n]$ satisfies

$$\texttt{cmp}(\&A[i],\&A[i+1]) < 0 \text{ for all } i = 1, 2, \ldots, n-1.$$

Quicksort. We are going to consider only one sorting algorithm, namely, one of the many variations on the theme of quicksort. Here it goes.

```
QUICKSORT($A$, first, last):
if   first < last
then mid = PARTITION($A$, first, last);
     QUICKSORT($A$, first, mid − 1);
     QUICKSORT($A$, mid + 1, last);
end
```

```
PARTITION($A$, first, last):
left $=$ first $+ 1$, right $=$ last;
repeat
        while  left $\leq$ last and cmp(&$A$[left],&$A$[first]) $< 0$
        do     left $=$ left $+ 1$;
        end

        while  cmp(&$A$[right],&$A$[first]) $> 0$
        do     right $=$ right $- 1$;
        end

        if     right $\leq$ left
        then   swap $A$[first] and $A$[right];
               return right;
        else   swap $A$[left] and $A$[right];
               left $=$ left $+ 1$, right $=$ right $- 1$;
        end
end
```

Average-case analysis of sorting algorithms. A typical analysis of a sorting algorithm is restricted to counting the number of calls of `cmp` and the number of record moves. For this purpose, the satellites are irrelevant and only the relative order of the keys matters: one may just as well assume that each record consists only of its key and that the key is an integer. Following this convention, we may

> replace the `cmp`(&A[left],&A[first]) < 0 in PARTITION
> by A[left]$<$ A[first]

and we may

> replace the `cmp`(&A[right],&A[first]) > 0 in PARTITION
> by A[right]$>$ A[first].

Furthermore, a common convention in the *average-case* analysis of a sorting algorithm is to assume that

($\star$) the n keys are the first n positive integers and
the average is taken over the n! distinct inputs.

Results of such average-case analysis may and may not describe the typical behavior of the sorting algorithm. In many applications, the input files are files that have been previously sorted and subsequently corrupted (consider the task of updating, every ten years, a list of all the U.S. municipalities sorted by population count); average-case analysis carried out under assumption ($\star$) is irrelevant to the behavior of sorting algorithms on such *nearly sorted* inputs.

PARTITION preserves randomness. Our average-case analysis of QUICKSORT hinges on a subtle property of our PARTITION. Let us illustrate this property on the 120 inputs with $n = 6$ and $A[1]$=3:

312456 $\to$ 213456	312465 $\to$ 213465	312546 $\to$ 213546	312564 $\to$ 213564
312645 $\to$ 213645	312654 $\to$ 213654	314256 $\to$ 213456	314265 $\to$ 213465
314526 $\to$ 213546	314562 $\to$ 213564	314625 $\to$ 213645	314652 $\to$ 213654
315246 $\to$ 213546	315264 $\to$ 213564	315426 $\to$ 213456	315462 $\to$ 213465
315624 $\to$ 213654	315642 $\to$ 213645	316245 $\to$ 213645	316254 $\to$ 213654
316425 $\to$ 213465	316452 $\to$ 213456	316524 $\to$ 213564	316542 $\to$ 213546
321456 $\to$ 123456	321465 $\to$ 123465	321546 $\to$ 123546	321564 $\to$ 123564
321645 $\to$ 123645	321654 $\to$ 123654	324156 $\to$ 123456	324165 $\to$ 123465
324516 $\to$ 123546	324561 $\to$ 123564	324615 $\to$ 123645	324651 $\to$ 123654
325146 $\to$ 123546	325164 $\to$ 123564	325416 $\to$ 123456	325461 $\to$ 123465
325614 $\to$ 123654	325641 $\to$ 123645	326145 $\to$ 123645	326154 $\to$ 123654
326415 $\to$ 123465	326451 $\to$ 123456	326514 $\to$ 123564	326541 $\to$ 123546
341256 $\to$ 123456	341265 $\to$ 123465	341526 $\to$ 123546	341562 $\to$ 123564
341625 $\to$ 123645	341652 $\to$ 123654	342156 $\to$ 213456	342165 $\to$ 213465
342516 $\to$ 213546	342561 $\to$ 213564	342615 $\to$ 213645	342651 $\to$ 213654
345126 $\to$ 123546	345162 $\to$ 123564	345216 $\to$ 213546	345261 $\to$ 213564
345612 $\to$ 123654	345621 $\to$ 213654	346125 $\to$ 123645	346152 $\to$ 123654
346215 $\to$ 213645	346251 $\to$ 213654	346512 $\to$ 123564	346521 $\to$ 213564
351246 $\to$ 123546	351264 $\to$ 123564	351426 $\to$ 123456	351462 $\to$ 123465
351624 $\to$ 123654	351642 $\to$ 123645	352146 $\to$ 213546	352164 $\to$ 213564
352416 $\to$ 213456	352461 $\to$ 213465	352614 $\to$ 213654	352641 $\to$ 213645
354126 $\to$ 123456	354162 $\to$ 123465	354216 $\to$ 213456	354261 $\to$ 213465
354612 $\to$ 123645	354621 $\to$ 213645	356124 $\to$ 123654	356142 $\to$ 123645
356214 $\to$ 213654	356241 $\to$ 213645	356412 $\to$ 123465	356421 $\to$ 213465
361245 $\to$ 123645	361254 $\to$ 123654	361425 $\to$ 123465	361452 $\to$ 123456
361524 $\to$ 123564	361542 $\to$ 123546	362145 $\to$ 213645	362154 $\to$ 213654
362415 $\to$ 213465	362451 $\to$ 213456	362514 $\to$ 213564	362541 $\to$ 213546
364125 $\to$ 123465	364152 $\to$ 123456	364215 $\to$ 213465	364251 $\to$ 213456
364512 $\to$ 123546	364521 $\to$ 213546	365124 $\to$ 123564	365142 $\to$ 123546
365214 $\to$ 213564	365241 $\to$ 213546	365412 $\to$ 123456	365421 $\to$ 213456

Each of the 120 outputs $A[1]A[2]A[3]A[4]A[5]A[6]$ has

$$A[1]{=}3 \text{ and } \{A[1],A[2]\} = \{1,2\} \text{ and } \{A[4],A[5],A[6]\} = \{4,5,6\},$$

as required of any variant of PARTITION; among these 120 outputs,

> each of the two permutations 12 and 21
> appears sixty times as $A[1]A[2]$

and

> each of the six permutations 456, 465, 546, 564, 645, 654
> appears twenty times as $A[4]A[5]A[6]$.

More generally, our PARTITION *preserves randomness* in the sense that, for every choice of positive integers n and k such that $1 \leq k \leq n$,

given the $(n-1)!$ inputs $A[1]A[2]\ldots A[n]$ with $A[1]{=}k$,
it produces $(n-1)!$ outputs such that

> each of the $(k-1)!$ permutations of $1, 2, \ldots, k-1$
> appears equally often as $A[1]A[2]\ldots A[k-1]$

and

> each of the $(n-k)!$ permutations of $k+1, k+2\ldots, n$
> appears equally often as $A[k+1]A[k+2]\ldots A[n]$.

We skip a justification of this claim.

2 Average-case analysis of $\textsc{Partition}(A, 1, n)$

2.1 The average number of key comparisons

Let us write $k = A[1]$. Cursor "left" moves from 2 to $k+1$ and cursor "right" moves from n to k. During this process, cursor "left" compares $A[1]$ with all of $A[2]$, $A[3]$, …, $A[n]$ in case $k = n$ and with all of $A[2]$, $A[3]$, …, $A[k+1]$ in case $k < n$; cursor "right" compares $A[1]$ with all of $A[n]$, $A[n-1]$,…, $A[k]$; the total number of key comparisons comes to n in case $k = n$ and to $n+1$ in case $k < n$. Since precisely $(n-1)!$ of the $n!$ inputs have $k = 1$, we conclude that

$$\textsc{Partition}(A, 1, n) \text{ makes } \left(n + 1 - \frac{1}{n}\right) \text{ key comparisons on the average.}$$

2.2 The average number of record swaps

For every choice of positive integers n and k such that $1 \leq k \leq n$, let $f(n, k)$ denote average number of times that $\textsc{Partition}(A, 1, n)$, given an input $A[1]A[2]\ldots A[n]$ with $A[1]{=}k$, executes the instruction

$$\text{swap } A[\text{left}] \text{ and } A[\text{right}];$$

In computing $f(n, k)$, let us call a key *small* if its value is less than k and let us call a key *large* if its value is more than k: in these terms, $f(n, k)$ equals the average number of small keys that appear in $A[k+1]A[k+2]\ldots A[n]$. Now imagine all the inputs $A[1]A[2]\ldots A[n]$ with $A[1]{=}k$ represented as the rows of an $r \times n$ matrix (of course, we have $r = (n-1)!$, but the value of r is irrelevant to our argument). In each column except the first (which holds k throughout), all of the $n-1$ keys distinct from k appear equally often; to put it differently, each of these keys appears $r/(n-1)$ times in every column except the first. In particular, the number of small entries in the last $n-k$ columns of the matrix is

$$(k-1) \cdot (n-k) \cdot \frac{r}{n-1}$$

and so

$$f(n,k) = \frac{(k-1)(n-k)}{n-1}$$

To express

$$\frac{1}{n}\sum_{k=1}^{n} f(n,k),$$

in a closed form, note that

$$\sum_{k=1}^{n}(k-1)(n-k) = \binom{n}{3}:$$

here, the left-hand side counts the number of ways to choose first an integer k from $\{1,2,\ldots n\}$, then an integer from $\{1,2,\ldots k-1\}$, and finally an integer from $\{k+1,k+2,\ldots n\}$, while the right-hand side counts the number of ways to choose three distinct integers from $\{1,2,\ldots n\}$. Hence

$$\frac{1}{n}\sum_{k=1}^{n} f(n,k) = \frac{n-2}{6}.$$

Recalling the swap of A[first] and A[right] that we have not accounted for yet, we conclude that

$$\textsc{Partition}(A,1,n) \text{ makes } \frac{n+4}{6} \text{ record swaps on the average.}$$

3 Average-case analysis of $\textsc{Quicksort}(A,1,n)$

Let $C(n)$ denote the average number of key comparisons made by $\textsc{Quicksort}(A,1,n)$ and let $S(n)$ denote the average number of record swaps made by $\textsc{Quicksort}(A,1,n)$. Trivially, we have

$$C(0) = C(1) = 0 \tag{1}$$

and

$$S(0) = S(1) = 0; \tag{2}$$

since our Partition preserves randomness, we have

$$C(n) = \left(n+1-\frac{1}{n}\right) + \frac{1}{n}\sum_{k=1}^{n} C(k-1) + \frac{1}{n}\sum_{k=1}^{n} C(n-k) \quad \text{whenever } n \geq 2 \tag{3}$$

and

$$S(n) = \frac{n+4}{6} + \frac{1}{n}\sum_{k=1}^{n} S(k-1) + \frac{1}{n}\sum_{k=1}^{n} S(n-k) \quad \text{whenever } n \geq 2. \tag{4}$$

The *recurrence relation* (3) expresses the n-th term of the sequence $C(0)$, $C(1)$, $C(2)$, $C(3)$, …as a function of the preceding terms, and so it, along with the *initial conditions* (1), determines the entire sequence: using (3) again and again, we find step by step

$$C(2) = 5/2,\quad C(3) = 16/3,\quad C(4) = 26/3,\quad C(5) = 62/5,\quad C(6) = 247/15,\quad \ldots$$

Similarly, the recurrence relation (4) expresses the n-th term of the sequence $S(0)$, $S(1)$, $S(2)$, $S(3)$, ... as a function of the preceding terms, and so it, along with the initial conditions (2), determines the entire sequence: using (4) again and again, we find step by step

$$S(2) = 1, \ \ S(3) = 11/6, \ \ S(4) = 11/4, \ \ S(5) = 56/15, \ \ S(6) = 859/180, \ \ \ldots$$

Solving a recurrence means expressing the n-th term of the sequence directly, without any references to the preceding terms. We are going to solve (3) with the initial conditions (1) and we are going to solve (4) with the initial conditions (2). These two procedures begin in similar ways, which we are going to treat as one in a more general setting: we shall solve the recurrence

$$f(n) = g(n) + \frac{1}{n}\sum_{k=1}^{n} f(k-1) + \frac{1}{n}\sum_{k=1}^{n} f(n-k) \ \text{ whenever } n \ge 2 \tag{5}$$

with the initial conditions

$$f(0) = f(1) = 0 \tag{6}$$

in terms of $g(2), g(3), g(4), \ldots$.

A cosmetic change. Observing that $\sum_{k=1}^{n} f(k-1) = \sum_{k=1}^{n} f(n-k)$, we record (5) as

$$f(n) = g(n) + \frac{2}{n}\sum_{k=1}^{n} f(k-1) \ \text{ whenever } n \ge 2. \tag{7}$$

A substantial change. An unpleasant feature of (7) is that it is a *full history* recurrence: each term $f(n)$ is a function of *all* the terms $f(1)$, $f(2)$,...,$f(n-1)$ preceding it. To convert (7) to a form where $f(n)$ is a function of $f(n-1)$ alone, let us substitute $n-1$ for n in (7):

$$f(n-1) = g(n-1) + \frac{2}{n-1}\sum_{i=0}^{n-2} f(i) \ \text{ whenever } n \ge 3. \tag{8}$$

If only the coefficient in front of the $\sum f(i)$ in (7) matched the coefficient in front of the $\sum f(i)$ in (8), we could make progress at once: subtracting (8) from (7) would yield an equation in $f(n)$ and $f(n-1)$ that involves no other $f(i)$. Fortunately, it is easy to make the "if only" come true: all we have to do is multiply (7) by n and multiply (8) by $n-1$. From

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

we subtract

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

obtaining

$$nf(n) - (n-1)f(n-1) = ng(n) + 2f(n-1) - (n-1)g(n-1);$$

after simplifications, we conclude that

$$f(n) = \frac{n+1}{n}f(n-1) + \Big(g(n) - \frac{n-1}{n}g(n-1)\Big) \ \text{ whenever } n \ge 3. \tag{9}$$

Telescoping. Recurrence (9) amounts to a sequence of identities

$$\begin{aligned}
f(3) &= \frac{4}{3}f(2) + \Big(g(3) - \frac{2}{3}g(2)\Big)\,, && (10)\\
f(4) &= \frac{5}{4}f(3) + \Big(g(4) - \frac{3}{4}g(3)\Big)\,, && (11)\\
f(5) &= \frac{6}{5}f(4) + \Big(g(5) - \frac{4}{5}g(4)\Big)\,, && (12)\\
\ldots &= \ldots\\
f(n-1) &= \frac{n}{n-1}f(n-2) + \Big(g(n-1) - \frac{n-2}{n-1}g(n-2)\Big)\,,\\
f(n) &= \frac{n+1}{n}f(n-1) + \Big(g(n) - \frac{n-1}{n}g(n-1)\Big)\,,\\
\ldots &= \ldots
\end{aligned}$$

If only the coefficient at $f(3)$ in (11) matched the coefficient at $f(3)$ in (10), and the coefficient at $f(4)$ in (12) matched the coefficient at $f(4)$ in (11), and so on, we could solve (9) at once: the sum of the first $n-2$ identities in this sequence would *telescope* through cancellations into a formula expressing $f(n)$ without references to any other $f(i)$ except $f(2)$. Fortunately, a simple trick makes the "if only" come true: all we have to do is divide (10) by 4, divide (10) by 5, and so on:

$$\begin{aligned}
\frac{f(3)}{4} &= \frac{f(2)}{3} + \frac{1}{4}\Big(g(3) - \frac{2}{3}g(2)\Big)\,,\\
\frac{f(4)}{5} &= \frac{f(3)}{4} + \frac{1}{5}\Big(g(4) - \frac{3}{4}g(3)\Big)\,,\\
\frac{f(5)}{6} &= \frac{f(4)}{5} + \frac{1}{6}\Big(g(5) - \frac{4}{5}g(4)\Big)\,,\\
\ldots &= \ldots\\
\frac{f(n-1)}{n} &= \frac{f(n-2)}{n-1} + \frac{1}{n}\Big(g(n-1) - \frac{n-2}{n-1}g(n-2)\Big)\,,\\
\frac{f(n)}{n+1} &= \frac{f(n-1)}{n} + \frac{1}{n+1}\Big(g(n) - \frac{n-1}{n}g(n-1)\Big)\,,\\
\ldots &= \ldots
\end{aligned}$$

The sum of the first $n-2$ identities in this sequence telescopes into

$$\frac{f(n)}{n+1} = \frac{f(2)}{3} - \frac{g(2)}{6} + \sum_{i=3}^{n-1} \frac{2g(i)}{(i+1)(i+2)} + \frac{g(n)}{n+1};$$

observing that $f(2) = g(2)$ by (6) and (7), we conclude that

$$f(n) = 2(n+1)\sum_{i=2}^{n-1} \frac{g(i)}{(i+1)(i+2)} + g(n) \quad \text{whenever } n \geq 3. \tag{13}$$

3.1 The average number of key comparisons

Setting $f(n) = C(n)$ and

$$g(n) = \left(n + 1 - \frac{1}{n}\right),$$

we reduce (6),(5) to (1),(3). It follows that

$$C(n) = 2(n+1)\sum_{i=2}^{n-1} \frac{i^2+i-1}{i(i+1)(i+2)} + \frac{n^2+n-1}{n} \quad \text{whenever } n \geq 3. \tag{14}$$

Partial fractions. To simplify the sum

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

we may use a trick from integral calculus, known as "partial fractions": if A, B, C are constants such that

$$\frac{i^2+i-1}{i(i+1)(i+2)} = \frac{A}{i} + \frac{B}{i+1} + \frac{C}{i+2} \quad \text{for all } i, \tag{15}$$

then

$$\begin{aligned} \sum_{i=2}^{n-1} \frac{i^2+i-1}{i(i+1)(i+2)} &= \sum_{i=2}^{n-1} \frac{A}{i} + \sum_{i=2}^{n-1} \frac{B}{i+1} + \sum_{i=2}^{n-1} \frac{C}{i+2} \\ &= A\sum_{i=2}^{n-1} \frac{1}{i} + B\sum_{i=3}^{n} \frac{1}{i} + C\sum_{i=4}^{n+1} \frac{1}{i} \\ &= \frac{5A}{6} + \frac{B}{3} + (A+B+C)\sum_{i=4}^{n-1} \frac{1}{i} + \frac{B}{n} + \frac{C(2n+1)}{n(n+1)} \end{aligned}$$

To find the values of A, B, C, we write (15) as

$$i^2 + i - 1 = (A+B+C)i^2 + (3A+2B+C)i + 2A$$

and require that $1 = A + B + C$, $1 = 3A + 2B + C$, $-1 = 2A$; this requirement is satisfied by $A = -1/2$, $B = 1$, $C = 1/2$; we conclude that

$$\sum_{i=2}^{n-1} \frac{i^2+i-1}{i(i+1)(i+2)} = \frac{-1}{12} + \sum_{i=4}^{n} \frac{1}{i} + \frac{2n+1}{2n(n+1)}. \tag{16}$$

Substituting from (16) into (14) we get, after simplifications,

$$C(n) = 2(n+1)\sum_{i=1}^{n} \frac{1}{i} - \frac{17n+5}{6}. \tag{17}$$

Harmonic numbers. It is not clear from (17) just how fast $C(n)$ grows with n. The second term is a simple linear function of n; it is the sum in the first term, whose order of magnitude is not obvious. It is customary to write

$$H_n = \sum_{i=1}^{n} \frac{1}{i} :$$

these numbers $H_1, H_2, H_3, \ldots$, being partial sums of the *harmonic series* $\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \ldots$, are known as the *harmonic numbers*. In this notation, (17) can be written as

$$C(n) = 2(n+1)H_n - \frac{17n+5}{6}. \tag{18}$$

How fast do the harmonic numbers grow? Since

$$\int_i^{i+1} \frac{1}{x}\,dx \le \frac{1}{i} \le \int_{i-1}^{i} \frac{1}{x}\,dx,$$

we have

$$H_n = 1 + \sum_{i=2}^{n} \frac{1}{i} \le 1 + \sum_{i=2}^{n} \int_{i-1}^{i} \frac{1}{x}\,dx = 1 + \int_1^n \frac{1}{x}\,dx = 1 + \ln n$$

and

$$H_n = \sum_{i=1}^{n-1} \frac{1}{i} + \frac{1}{n} \ge \sum_{i=1}^{n-1} \int_i^{i+1} \frac{1}{x}\,dx + \frac{1}{n} = \int_1^n \frac{1}{x}\,dx + \frac{1}{n} = \ln n + \frac{1}{n};$$

hence

$$\frac{1}{n} \le H_n - \ln n \le 1 \quad \text{for all } n. \tag{19}$$

Substituting into (18), we conclude that

$$C(n) = 2n\ln n + \alpha(n) \text{ with } |\,\alpha(n)| \in O(n).$$

Euler's constant. By the way, since each of the integrals in the right-hand side of

$$\ln n - (H_n - 1) = \sum_{i=2}^{n} \int_{i-1}^{i} \left(\frac{1}{x} - \frac{1}{i}\right) dx,$$

is positive, the differences $\ln n - H_n$ grow with n. To put it differently, the differences $H_n - \ln n$ decrease as n grows; since these differences are sandwiched between 0 and 1 by (19), it follows that

$$\lim_{n\to\infty} (H_n - \ln n)$$

exists. This limit is known as *Euler's constant* and denoted γ; like π and e, it is one of the more notorious irrational numbers; its value is about 0.577.

3.2 The average number of record swaps

Setting $f(n) = S(n)$ and

$$g(n) = \frac{n+4}{6},$$

we reduce (6),(5) to (2),(4). It follows that

$$S(n) = \frac{n+1}{3} \sum_{i=2}^{n-1} \frac{i+4}{(i+1)(i+2)} + \frac{n+4}{6} \quad \text{whenever } n \geq 3. \tag{20}$$

We have

$$\begin{aligned} \sum_{i=2}^{n-1} \frac{i+4}{(i+1)(i+2)} &= \sum_{i=2}^{n-1} \frac{3}{i+1} - \sum_{i=2}^{n-1} \frac{2}{i+2} = 3\sum_{i=3}^{n} \frac{1}{i} - 2\sum_{i=4}^{n+1} \frac{1}{i} \\ = \quad & 1 + \sum_{i=4}^{n} \frac{1}{i} - \frac{2}{n+1} = H_n - \frac{5}{6} - \frac{2}{n+1}; \end{aligned}$$

substituting into (14), we get

$$S(n) = \frac{n+1}{3} H_n - \frac{2n+5}{18} \quad \text{whenever } n \geq 3;$$

appealing to (19),we conclude that

$$S(n) = \tfrac{1}{3} n \ln n + \beta(n) \text{ with } |\,\beta(n)| \in O(n).$$

_____*****_____

These notes are based on

R. Sedgewick "The Analysis of Quicksort Programs", *Acta Informatica* **7**, 327-355, 1977.