

# 1 Quicksort

---

## 1.1 Pseudocode

---

**Algorithm 1** Quicksort

---

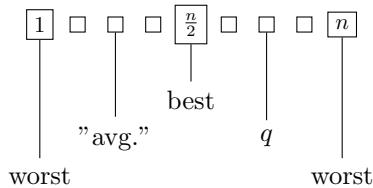
```
1: procedure QUICKSORT( $A, p, r$ )
2:   if  $p < r$  then
3:      $q \leftarrow \text{PARTITION}(A, p, r)$ 
4:     QUICKSORT( $A, p, q - 1$ )
5:     QUICKSORT( $A, q + 1, r$ )
6:   end if
7: end procedure

8: function PARTITION( $A, p, r$ )
9:    $X \leftarrow A[r]$ 
10:   $i \leftarrow p - 1$ 
11:  for  $j = p$  to  $r - 1$  do
12:    if  $A[j] \leq X$  then
13:       $i \leftarrow i + 1$ 
14:       $A[i] \leftrightarrow A[j]$ 
15:    end if
16:  end for
17:   $A[i + 1] \leftrightarrow A[r]$ 
18:  return  $(i + 1)$ 
19: end function
```

---

## 1.2 Analysis

The worst case is when the pivot is either in the *first* or *last* index, best case is when the pivot is the *median* index, and the "average" case is when the pivot is between the first/last and median index (here  $q$  will represent the *true* average).



### Worst Case

As a recurrence,

$$\begin{aligned} T(n) &= T(n-1) + n - 1, \quad T(0) = T(1) = 0 \\ &= \sum_{i=1}^{n-1} i = \frac{(n-1)n}{2} \end{aligned}$$

### Best Case

As a recurrence,

$$\begin{aligned} T(n) &= 2T\left(\frac{n}{2}\right) + n - 1, \quad T(0) = T(1) = 0 \\ &= n \lg n - n + 1 \end{aligned}$$

Note: This is the same recurrence as Mergesort!

### Average Case

As a recurrence,

$$T(n) = T\left(\frac{n}{4}\right) + T\left(\frac{3n}{4}\right) + n - 1, \quad T(0) = T(1) = 0$$

Solve with Strong Constructive Induction.

Guess:  $T(n) \leq an \lg n$  for  $n \geq 1$  and some constant  $a$ .

Base case  $n = 1$ :  $an \lg n = a1 \lg 1 = a \cdot 1 \cdot 0 = 0$ ,  $T(1) = 0$  and  $0 \leq 0$ .

Inductive Hypothesis: Assume true for  $n < k$ ,  $T(k) \leq ak \lg k$  for  $1 \leq k \leq n$ .

Inductive Step:

$$\begin{aligned} T(n) &= T\left(\frac{n}{4}\right) + T\left(\frac{3n}{4}\right) + n - 1 \\ &\leq a\frac{n}{4} \lg\left(\frac{n}{4}\right) + a\frac{3n}{4} \lg\left(\frac{3n}{4}\right) + n - 1 \quad \text{by IH} \\ &\leq a\frac{n}{4} (\lg n - \lg 4) + a\frac{3n}{4} (\lg 3 + \lg n - \lg 4) + n - 1 \\ &\leq \frac{an}{4} (\lg n - 2) + \frac{3an}{4} (\lg 3 + \lg n - 2) + n - 1 \\ &\leq \frac{an \lg n}{4} - \frac{2an}{4} + \frac{3an \lg 3}{4} + \frac{3an \lg n}{4} - \frac{2 \cdot 3an}{4} + n - 1 \\ &\leq an \lg n + \left(-\frac{a}{2} + \frac{3a \lg 3}{4} - \frac{3}{2}a + 1\right) n - 1 \\ &\leq an \lg n + \left(\left(\frac{3 \lg 3}{4} - 2\right)a + 1\right) n - 1 \end{aligned}$$

It follows that we need

$$\left(\frac{3\lg 3}{4} - 2\right)a + 1 \leq 0 \implies a \geq \frac{1}{2 - \frac{3\lg 3}{4}}$$

$$a \gtrsim 1.23$$

Thus,

$$T(n) \lesssim 1.23n \lg n$$

### Exact Average Case

As a recurrence,

$$\begin{aligned} T(n) &= \left[ \sum_{q=1}^n \frac{1}{n} (T(q-1) + T(n-q)) \right] + n - 1, \quad T(0) = T(1) = 0 \\ &= \frac{1}{n} \sum_{q=1}^n (T(q-1) + T(n-q)) + n - 1 \\ &= \frac{1}{n} \sum_{q=1}^n T(q-1) + \frac{1}{n} \sum_{q=1}^n T(n-q) + n - 1 \\ &= \frac{1}{n} \sum_{q=0}^{n-1} T(q) + \frac{1}{n} \sum_{q=0}^{n-1} T(n-q) + n - 1^* \\ &= \frac{2}{n} \sum_{q=0}^{n-1} T(q) + n - 1 \end{aligned}$$

\*Note that:

$$\begin{aligned} \sum_{q=1}^n T(q-1) &= T(0) + T(1) + T(2) + \cdots + T(n-1) \\ \sum_{q=1}^n T(n-q) &= T(n-1) + T(n-2) + T(n-3) + \cdots + T(0) \end{aligned}$$

Solve with Strong Constructive Induction.

Guess:  $T(n) \leq an \lg n$  for  $n \geq 1$  and some constant  $a$ .

Base case  $n = 1$ :  $an \lg n = a1 \lg 1 = a \cdot 1 \cdot 0 = 0$ ,  $T(1) = 0$  and  $0 \leq 0$ .

Inductive Hypothesis: Assume true for  $n$ ,  $T(k) \leq ak \lg k$  for  $1 \leq k \leq n$ .

Inductive Step:

$$\begin{aligned}
T(n) &= n - 1 + \frac{2}{n} \sum_{q=0}^{n-1} T(q) \\
&= n - 1 + \frac{2}{n} \sum_{q=1}^{n-1} T(q) \\
&\leq n - 1 + \frac{2}{n} \sum_{q=1}^{n-1} aq \lg q \quad \text{by IH} \\
&\leq n - 1 + \frac{2a}{n} \int_1^n x \lg x dx \quad \text{by integral bound} \\
&= n - 1 + \frac{2a}{n} \left[ \frac{x^2 \lg x}{2} - \frac{x^2 \lg e}{4} \right] \Big|_1^n \\
&= n - 1 + an \lg n - \frac{an \lg e}{2} + \frac{a \lg e}{2n} \\
&= an \lg n + \left[ 1 - \frac{a \lg e}{2} \right] n - 1 + \frac{a \lg e}{2n}
\end{aligned}$$

It follows that we need

$$1 - \frac{a \lg e}{2} \leq 0 \implies a \geq \frac{2}{\lg e}$$

So set  $a = \frac{2}{\lg e} \approx 1.39$ , we then need

$$\frac{a \lg e}{2n} - 1 \leq 0 \iff \frac{2 \lg e}{(\lg e) 2n} - 1 \leq 0 \iff \frac{1}{n} - 1 \leq 0$$

Thus,

$$T(n) \lesssim 1.39n \lg n$$

We can realize a more natural formula,

$$T(n) \leq an \lg n = \frac{2n \lg n}{\lg e} = 2n \ln n$$

### Theorem

*The expected number of comparisons for Quicksort is  $\approx 2n \ln n$ .*

### Blum's Exact Average Case

Let  $P(i, j)$  be the probability that the  $i$ th smallest and  $j$ th smallest elements are compared.

### Theorem

The average number of comparisons for quicksort is

$$\sum_{1 \leq i < j \leq n} P(i, j)$$

Only matters the first time an element from  $A[i], \dots, A[j]$  is picked as pivot. The probability that the  $i$ th and  $j$ th smallest elements are compared during the execution of quicksort is:

$$\frac{2}{j - i + 1}$$

Then,

$$\begin{aligned} \sum_{1 \leq i < j \leq n} P(i, j) &= \sum_{i=1}^n \sum_{j=i+1}^n P(i, j) = \sum_{i=1}^n \sum_{j=i+1}^n \frac{2}{j - i + 1} \\ &= 2 \sum_{i=1}^n \sum_{j=i+1}^n \frac{1}{j - i + 1} \\ &= 2 \sum_{i=1}^n \left( \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n-i+1} \right) \\ &= 2 \sum_{i=1}^n (H_{n-i+1} - 1) = 2 \sum_{i=1}^n H_{n-i+1} - 2 \sum_{i=1}^n 1 \\ &= 2(H_n + H_{n-1} + \dots + H_1) - 2 \sum_{i=1}^n 1 \\ &= 2 \sum_{i=1}^n H_i - 2n \\ &= 2((n+1)H_n - n) - 2n \\ &= 2(n+1)H_n - 4n \approx 2n \ln n \end{aligned}$$

## 1.3 Comments

### Quicksort Code

Recall the Quicksort procedure,

```
procedure QUICKSORT( $A, p, r$ )
if  $p < r$  then
     $q \leftarrow \text{PARTITION}(A, p, r)$ 
    QUICKSORT( $A, p, q - 1$ )
    QUICKSORT( $A, q + 1, r$ )
end if
end procedure
```

When executing  $\text{quicksort}(A, p, q - 1)$  we must stack up  $\text{quicksort}(A, q + 1, r)$  to be executed later. We need to store the pair of index values  $(q + 1, r)$ .

## Stack in Action

The left table represents the pivot being the largest element, and the right table represents the pivot being the smallest element.

Pivot	Stack	Pivot	Stack
$A[n]$	$(n+1, n)$	$A[1]$	$(2, n)$
$A[n-1]$	$(n, n-1)$	$A[2]$	$(3, n)$
$A[n-2]$	$(n-1, n-2)$	$A[3]$	$(4, n)$
:	:	:	:
$A[3]$	$(4, 3)$	$A[n-2]$	$(n-1, n)$
$A[2]$	$(3, 2)$	$A[n-1]$	$(n, n)$
$A[1]$		$A[n]$	

When the pivot is the smallest element the stack remains very small but if the pivot is the largest element we stack  $n - 1$  problems to do later.

## In Place

On average, height of the stack for Quicksort is  $\Theta(\log n)$ . Height of stack for Mergesort is  $\lg n + O(1)$ .

**Definition:** An algorithm is *in place* if it uses  $O(1)$  extra variables and (on average)  $O(\log n)$  extra index variables.

A more technical definition is; An algorithm is *in place* if it uses at most  $O(1)$  extra variables and (on average)  $O((\log n)^2)$  extra bits.

## Stack

We can modify the Quicksort procedure so that stack has height at most  $\lg n$ .

```

procedure QUICKSORT( $A, p, r$ )
  if  $p < r$  then
     $q \leftarrow \text{PARTITION}(A, p, r)$ 
    if  $q \leq (p+r)/2$  then
      QUICKSORT( $A, p, q-1$ )
      QUICKSORT( $A, q+1, r$ )
    else
      QUICKSORT( $A, q+1, r$ )
      QUICKSORT( $A, p, q-1$ )
    end if
  end if
end procedure

```

### **Is Quicksort In Place?**

Quicksort is in place, but it does not use a constant amount of extra space. Quicksort uses slightly more than a constant amount of extra space.

### **Choice of Pivots**

It is risky to pivot on the last element because the last element could be the largest element. Some better ways could be;

- Pivot on middle element.
- Pivot on median of first, middle, and last elements.
- Pivot on random element.
- Pivot on median of three random elements.
- Pivot on median of five random elements. (Law of diminishing returns.)
- Randomly permute array before starting. (Equivalent to pivoting on random element.)

### **Partitioning**

There are a variety of partition routines. The current edition of the textbook has a version that uses  $n - 1$  comparisons, but the previous edition of the textbook has a version which uses  $n$  comparisons.