1. The Quicksort Algorithm

Robert R. Snapp
snapp@cs.uvm.edu

Department of Computer Science
University of Vermont

Last modified on April 26, 2012
Outline

1. The Algorithm
2. Complexity Analysis
3. Summing Lemmas
   - Harmonic Numbers $H_n$
4. Expected Number of Comparisons
5. Empirical Comparison
6. Bibliography
The quicksort algorithm

**PROBLEM:** Given a sequence of numbers,

\[(x_1, x_2, \ldots, x_n),\]

find a permutation \(\pi_1, \pi_2, \ldots, \pi_n\) of \(\{1, 2, \ldots, n\}\) such that \(x_{\pi_i} \leq x_{\pi_j}\) for all \(i < j\), making

\[(x_{\pi_1}, x_{\pi_2}, \ldots, x_{\pi_n})\]

an increasing-ordered list.

For example, \((24, 86, 12, 33)\) becomes \((12, 24, 33, 86)\).

Hoare’s [2] 1962 *quicksort* algorithm elegantly performs this trick recursively. In Haskell, for example:

```haskell
qsort []   = []
qsort (x:xs) = qsort [y | y <- xs, y < x] ++ [x] ++
                 qsort [y | y <- xs, y >= x]
```

Robert R. Snapp © 2011
(defn split
  "Given a sequence of numbers s (arg1) and a pivot value (arg2), split
partitions the sequence into two lists: the first being the elements that are
less than pivot, and the second being the elements greater than the pivot."
  [s pivot]
  (loop [bottom () top () s s]
    (if (empty? s)
      (vector bottom top)
      (let [[item & r] s]
        (if (< item pivot) ; a comparison
          (recur (conj bottom item) top r)
          (recur bottom (conj top item) r))))))

(defn qsort
  "Given a sequence of numbers, qsort returns a sorted version of the input
sequence in increasing order using Hoare’s quicksort algorithm."
  [[pivot & xs]]
  (when pivot
    (let [[bot top] (split xs pivot)]
      (concat (qsort bot) (list pivot) (qsort top))))
Complexity analysis

The worst-case performance of quicksort occurs if the elements in input list are ordered monotonically. In this case every element is eventually compared with every other one, resulting in a total of

\[
\binom{n}{2} = \frac{n(n - 1)}{2} = \Theta(n^2)
\]

comparisons. The number of comparisons is minimized if pivots can be chosen so that each sublist is evenly partitioned at each level of the algorithm. In this case the number of comparisons \( C(n) \) satisfies the recurrence relation:

\[
C(n) \leq 2C(n/2) + n - 1, \text{ with } C(0) = C(1) = 0.
\]

Whence \( C(n) \leq n \log_2 n + n = \Theta(n \ln n) \) [1]. Unfortunately, it is difficult to select pivots in this way without clairvoyance. However, if either the pivots are chosen at random, or if the order of the input list is randomized, then using probability theory we will show that the expected number of comparisons is reduced to

\[
\mathbb{E}\{C(n)\} = 2n \ln n + \Theta(n) = \Theta(n \ln n).
\]
Lemma (Triangular sums)

\[ \sum_{1 \leq i < j \leq n} a_{i,j} \overset{\text{def}}{=} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} a_{i,j} = \sum_{j=2}^{n} \sum_{i=1}^{j-1} a_{i,j}. \]

Proof (by picture):

![Diagram showing triangular sums](image-url)
Lemma (Triangular sums)

\[
\sum_{1 \leq i < j \leq n} a_{i,j} \overset{\text{def}}{=} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} a_{i,j} = \sum_{j=2}^{n} \sum_{i=1}^{j-1} a_{i,j}.
\]

Proof (by picture):

![Diagram showing the triangular sum calculation with arrows pointing to the corresponding elements in a matrix.](attachment:image.png)
Lemma (Triangular sums)

\[
\sum_{1 \leq i < j \leq n} a_{i,j} \overset{\text{def}}{=} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} a_{i,j} = \sum_{j=2}^{n} \sum_{i=1}^{j-1} a_{i,j}.
\]

Proof (by picture):
Lemma (Triangular sums (alternate form))

\[
\sum_{i=1}^{n-1} \sum_{k=2}^{n+1-i} a_{i,k} = \sum_{k=2}^{n} \sum_{i=1}^{n+1-k} a_{i,k}.
\]

Proof (by picture):
Lemma (Triangular sums (alternate form))

\[
\sum_{i=1}^{n-1} \sum_{k=2}^{n+1-i} a_{i,k} = \sum_{k=2}^{n} \sum_{i=1}^{n+1-k} a_{i,k}.
\]

Proof (by picture):

![Diagram showing the triangular sums](image-url)
Lemma (Triangular sums (alternate form))

\[
\sum_{i=1}^{n-1} \sum_{k=2}^{n+1-i} a_{i,k} = \sum_{k=2}^{n} \sum_{i=1}^{n+1-k} a_{i,k}.
\]

Proof (by picture):

[Diagram showing the sums with arrows and numbers representing the indices and values of \(a_{i,k}\).]
The $n$-th harmonic number is defined by

$$H_n = \sum_{k=1}^{n} \frac{1}{k}.$$ 

For example, $H_1 = 1$, $H_2 = 1 + \frac{1}{2} = \frac{3}{2}$, $H_3 = \frac{11}{6}$, etc.
Lemma (Bounds on the Harmonic numbers, $H_n$)

For $n \geq 1$,

$$\ln(n + 1) \leq H_n \leq 1 + \ln n.$$ 

Proof (by picture):

$$\int_0^n \frac{dx}{x + 1} \leq H_n \leq 1 + \int_1^n \frac{dx}{x}.$$
Lemma (Asymptotic approximation of $H_n$)

\[ H_n = \sum_{k=1}^{n} \frac{1}{k} \sim \gamma + \ln n + \frac{1}{2n} - \frac{1}{12n^2} + \frac{1}{120n^4} + O(n^{-6}), \]

where $\gamma = \int_{1}^{\infty} \left( \frac{1}{[x]} - \frac{1}{x} \right) \, dx \approx 0.57721 \cdots$ is the Euler-Mascheroni constant.

The (♣♣♣) proof, based on Euler’s summation formula

\[
\sum_{k=a}^{b} f(k) = \int_{a}^{b} f(x) \, dx + \sum_{k=1}^{m} \frac{B_k}{k!} f^{(k-1)}(x)|_{a}^{b} + R_m,
\]

where $B_k$ is the $k$-th Bernoulli number, and $R_m = \frac{(-1)^{m+1}}{m!} \int_{a}^{b} B_m(x - [x]) f^{(m)}(x) \, dx$ is a remainder term, can be found in Chapter 9 of *Concrete Mathematics* by Ronald L. Graham, Donald E. Knuth, and Oren Patashnik, Addison-Wesley, Reading, MA, 1994.
Theorem (Expected number of comparisons by quicksort)

The expected number of comparisons used by quicksort, when applied to a list of \( n \) numbers, is

\[
\mathbb{E}\{C(n)\} = 2(n + 1) H_n - 4n = 2n \log n + \Theta(n).
\]

Proof: Let \( X_{i,j} = 1 \) if at some point in the course of the algorithm \( x_{\pi_i} \) is compared to \( x_{\pi_j} \); otherwise \( X_{i,j} = 0 \). The total number of comparisons is then given by

\[
C(n) = \sum_{i<j} X_{i,j} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{i,j}.
\]

(Observe that in the worst case \( X_{i,j} \) always equals 1, so that \( C(n) = \binom{n}{2} \), as mentioned earlier.) Using the linearity of expectation,

\[
\mathbb{E}\{X\} = \mathbb{E}\left\{ \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{i,j} \right\} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathbb{E}\{X_{i,j}\}.
\]
Since $X_{i,j}$ is a binary (or indicator) random variable, $E\{X_{i,j}\}$ equals the probability that $x_{\pi_i}$ is at some time compared to $x_{\pi_j}$. This can only happen if one of them is the first pivot chosen from the set \{\(x_{\pi_i}, x_{\pi_{i+1}}, \ldots, x_{\pi_j}\)\}. If we assume pivots are selected at random, then

$$E\{X_{i,j}\} = \frac{2}{j - i + 1}.$$  

Thus,

$$E\{X\} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j - i + 1},$$

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

$$= \sum_{k=2}^{n} \sum_{i=1}^{n-1-k} \frac{2}{k},$$

$$= \sum_{k=2}^{n} (n+1-k) \frac{2}{k} = 2(n+1) \sum_{k=1}^{n} \frac{1}{k} - 4n$$

$$= 2(n+1) H_n - 4n$$
Histogram: number of comparisons
Simulation results

Quicksort estimates with 100 trials

\[ \binom{n}{2} \]

\[ 2n(n + 1)H_n - 4n \]

Sequence Length (n) vs. Expected Number of Comparisons
Bibliography I
