算法设计与分析: 作业 #3

201928015059003
杨天健

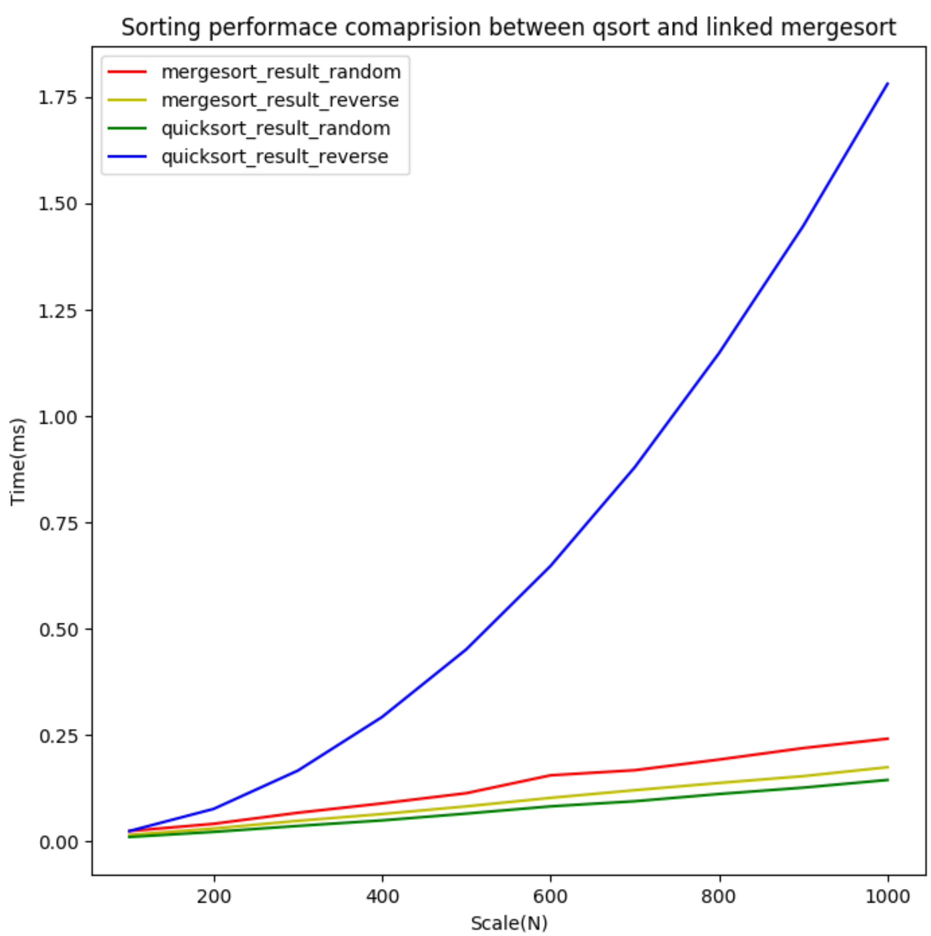
2019 年 9 月 30 日

Question 1

I've uploaded all of my code to `https://github.com/yangtianjian/Typical_algorithms`. The core code is in simple_algorithm/Sorting.hpp. In main function, main.cpp, you can uncomment the two lines: test_mergesort and test_quicksort. Don't forget to modity the parameter 'PATH' to fit for your local environment.

Question 2

I've made a comparison between linked merge sort and quicksort. For each algorithm, there are two cases. For one case, data are provided in random permutation. For another, it is given reversely. The result is provided below and **the time is given in milliseconds**:



Question 3

Whether we use an auxiliary array or a linked list, merge sort cannot be finished in-place. Another n-size of space is needed. Let's use $f(n)$ to indicate global variables, $g(n)$ for local variables and $T(n)$ for total space complexity. We have:

$$
\begin{aligned}
&f(n) = 2n = \Theta(n) \\
&g(n) = 2g(\frac{n}{2}) + \Theta(1) \Rightarrow g(n) = \Theta(n)
\end{aligned}
$$

So $T(n) = f(n) + g(n) = \Theta(n)$.

Question 4

Now we use induction method to prove the formula in the question that:

$$R[n] \leq 4cn \tag{1}$$

The precondition is (Here we get rid of serveral unnecessary notations to make it more clearly),

$$
\begin{aligned}
C[n][k] &\leq cn + \frac{1}{n}\left(\sum_{1\leq i<k} C[n-i][k-i] + \sum_{k<i\leq n} C[i-1][k]\right) && (2)\\
R[n] &= max_k(C[n][k]) && (3)\\
c &\geq R[1] && (4)
\end{aligned}
$$

To begin with, for $n = 1$, $R[1] \leq c \leq 4c$, it is true for (1).
Now we have $R[p] \leq 4cp$ for each $p = 1, 2, 3, ..., n-1$. And we start to prove.
Because $R[n] = max_k(C[n][k])$, for every k, $C[n][k] \leq R[n]$. So (2) can be written as:

$$
\begin{aligned}
C[n][k] &\leq cn + \frac{1}{n}\left(\sum_{1\leq i<k} C[n-i][k-i] + \sum_{k<i\leq n} C[i-1][k]\right) && (5)\\
&\leq cn + \frac{1}{n}\left(\sum_{1\leq i<k} R[n-i] + \sum_{k<i\leq n} R[i-1]\right) && (6)
\end{aligned}
$$

For each $1 \leq i < k$, $n - i \leq n - 1$. And for each $k < i \leq n$, $i - 1 \leq n - 1$. By using former iteration of induction, we have $R[n-i] \leq 4c(n-i)$ and $R[i-1] \leq 4c(i-1)$. From(6), we come to:

$$
\begin{aligned}
&cn + \frac{1}{n}\left(\sum_{1\leq i<k} R[n-i] + \sum_{k<i\leq n} R[i-1]\right) && (7)\\
&\leq cn + \frac{1}{n}\left(\sum_{1\leq i<k} 4c(n-i) + \sum_{k<i\leq n} 4c(i-1)\right) && (8)\\
&= cn + \frac{2c}{n}\Big[(2n-k)(k-1) + (k+n-1)(n-k)\Big] && (9)
\end{aligned}
$$

In the bracket is a quadratic polynomial respect to k. When $\frac{n+1}{2}$, the formula can be written as:

$$
\begin{aligned}
& cn + \frac{2c}{n}\left[(2n-k)(k-1) + (k+n-1)(n-k)\right] && (10)\\
& \leq cn + \frac{2c}{n}\left[\frac{1}{2}(n+1)^2 + n^2\right] - 6c && (11)\\
& \leq cn + \frac{2c}{n} * \frac{3}{2}n^2 - 6c && (12)\\
& = 4cn + (\frac{2}{n} - 4)c && (13)\\
& \leq 4cn && (14)
\end{aligned}
$$

$C[n][k] \leq 4cn$ for each $k = 1, 2, ..., p-1$, So we have $R[n] = max_k(C[n][k]) \leq 4cn$. So the problem is solved.