Rajshahi University of Engineering & Technology

CSE 2202: Sessional Based on CSE 2201

Lab Report 02

Date: November 13, 2018

Submitted to

Biprodip Pal

Instructor, CSE 2101 & CSE 2202

Assistant Professor, Dept. of CSE

Submitted by

Fuad Al Abir

Roll: 1603021

Section: A

Dept. of CSE

Sessional – Cycle 1 – Problem A

Efficiency consideration in terms of computation time for Bubble Sort and Merge Sort.

Machine Configuration:

Processor: Intel® Core™ i5-7299U CPU @ 2.50GHz 2.71 GHz

RAM: 4.00 GB (3.90 GB usable)

System Type: 64-bit Operating System, x64-based processor

Theoritical Complexity:

(i) Bubble Sort: $O(n^2)$

(ii) Merge Sort: $O(n \log n)$

Bubble Sort:

$$\sum_{i=0}^{n-1}\sum_{j=i+1}^{n} 1 = \sum_{i=0}^{n-1}(n-(i+1)-1) = \sum_{i=0}^{n-1} n-i = \sum_{i=0}^{n-1} n - \sum_{i=0}^{n-1} i = n^2 - \frac{n(n+1)}{2}$$
$$= \frac{2n^2-n^2-n}{2} = \frac{n^2-n}{2} \equiv O(n^2)$$

Merge Sort:

$$T(n) = \begin{cases} 1 & n=1 \\ 2T(\frac{n}{2})+cn & n>1 \end{cases}$$

$$T(n) = 2T(\frac{n}{2})+cn = 2\{2T(\frac{n}{4})+c\frac{n}{2}\}+cn = 2^2T(\frac{n}{2^2})+2cn$$
$$= 2^2\{2T(\frac{n}{8})+c\frac{n}{4}\}+2cn = 2^3T(\frac{n}{2^3})+3cn$$
$$= 2^kT(\frac{n}{2^k})+kcn \qquad \{\text{After } k^{th} \text{ iteration}\}$$

Let $k = log_2 n$,

$$T(n) = nT(\frac{n}{n})+log_2 n*cn = nT(1)+log_2 n*cn = n+log_2 n*cn \equiv O(nlog_2 n)$$

Input Size - N	Bubble Sort (in seconds)	Merge Sort (in seconds)
10000	0.287	0.002
50000	7.544	0.014
100000	24.579	0.021
250000	116.206	0.049

Table: Experimental Result (Time Requirement for Bubble and Merge Sort)

Sessional – Cycle 2 – Problem A

Efficiency consideration in terms of comparisons for maximum and minimum finding from a set of elements using brute force technique and divide conquer approach.

Machine Configuration:

Processor:	Intel® Core™ i5-7299U CPU @ 2.50GHz 2.71 GHz
RAM:	4.00 GB (3.90 GB usable)
System Type:	64-bit Operating System, x64-based processor

Theoritical Complexity:

(i) Brute force: O(n)

(ii) Divide and Conquer with Single Node Base: O(n)

(iii) Divide and Conquer with Single and Double Node Base: O(n)

Brute Force:

$$\sum_{i=1}^{n} 2 = 2n - 2 \equiv O(n)$$

Divide and Conquer with Single Node Base:

$$T(n) = \begin{cases} 0, & n = 1 \\ 2T(\frac{n}{2}) + c, & n > 1 \end{cases}$$

$$T(n) = 2T(\frac{n}{2}) + c = 2\{2T(\frac{n}{4}) + c\} + c = 2^2T(\frac{n}{2^2}) + 3c = 2^2T(\frac{n}{2^2}) + (2^2 - 1)c$$
$$= 2^2\{2T(\frac{n}{2^3}) + (2^3 - 1)c$$

After k^{th} iteration,

$$T(n) = 2^kT(\frac{n}{2^k}) + (2^k - 1)c \tag{1}$$

Let, $\frac{n}{2^k} = 1 \implies 2^k = n$ and $c = 2$

$$T(n) = nT(1) + (n-1)2 = n.0 + 2n - 2 = \mathbf{2n - 2} \equiv O(n)$$

Divide and Conquer with Single and Double Node Base:

$$T(n) = \begin{cases} 0, & n = 1 \\ 1, & n = 2 \\ 2T(\frac{n}{2}) + c, & n > 1 \end{cases}$$

Using equation (1),

Let, $\frac{n}{2^k} = 2 \implies 2^k = \frac{n}{2}$ and $c = 2$

$$T(n) = \frac{n}{2}T(2) + (\frac{n}{2} - 1)2 = \frac{n}{2} + \frac{2n}{2} - 2 = \mathbf{\frac{3n}{2} - 2} \equiv O(n)$$

K	Brute force (2K – 2)	Divide and Conquer (Case 1)	Divide and Conquer (Case 2)
10000	19998	19998	15902
50000	99998	99998	82766
100000	199998	199998	165534
250000	499998	499998	381070

Table: Experimental Result (Comparisons)