CP 312 A2

Liang Liu
153182750

1. Description of Algorithm:

use merge sort sort array A and array B to make them in order.
Then two for loops, one is inner, use to go through each element and to get sum of two elements in B and store them to a array called $\boxed{S}$.
Then also need two loops, one is inner. The outter loop loop each elements in $\boxed{S}$, inner go through each element in A, then use Binary Search to search if there's a result in A that equal to [sum of two elements in B] mins [one element in A]
Then if there's a result, store it as $\boxed{r}$ and use two loops to go through B again to find the right index of two elements in B that equal to $\boxed{r}$, then return each index in A and B

include sum of itself, ep: B[0]+B[0]

Pseudocode:

```
Mergesort(A);     S[];     length=0;   Mergesort(B);
for(i=0; i<n, i++)
    for(j=0; j<n, j++)
        S[ ] = B[i]+B[j]
        length++

od

for i=0; i<length; i++
    for j=0; j<n; j++
        k= BinarySearch(S[i]-A[j], A)
        if k>0 then
            Break
    od
od
for a=0; a<n; a++
    for b=0; b<n; b++
        if B[a]+B[b] == S[i] then
            i1 = j
            i2 = k
            j1 = a
            j2 = b
        return (i1, i2, j1, j2)
        end
```

Justification:

$A = [8, 9, 2]$

$B = [6, 3, 7]$

After sort: $A = [2, 8, 9]$

$B = [3, 6, 7]$

After first two loops:

After sort: $S[\,] = [6, 9, 10, 12, 13, 14]$

After second two loops:

$K = K = \text{BinarySearch}(S[2] - A[0], A)$

$\Downarrow$

$8 = A[K] = BS(10 - 2, A)$

After third two loops:

$B[0] + B[2] = 3 + 7 = 10$

So $i_1 = 0$ $i_2 = 1$ $j_1 = 0$ $j_2 = 2$

So the Algorithm corret.

Time complexity

mergesort's time complexity is $n\log n$

First two loops complexity $= n^2$

second two loops complexity $= n^2 \log n$

thrid two loops complexity $= n^2$

$\therefore n\log n + n^2 + n^2 \log n + n^2 = n^2 \log n$

Thum, this algorithm has $T(n) = \Theta(n^2 \log n)$

2. m := 1; s := 1;

while m <= n do

for j = 1 to $2\lceil \log m \rceil$ do

s := s+1

od

m := 3 * m

od

$$\sum_{m=1}^{\log_3^n} 2\log m \geq \sum_{m=\frac{\log_3^n}{2}+1}^{\log_3^n} 2\log m \geq \sum_{m=\frac{\log_3^n}{2}+1}^{\log_3^n} 2\log \frac{\log_3^n}{2}$$

$$\geq \frac{\log_3^n}{2} \cdot 2 \cdot \log \frac{\log_3^n}{2} \in \Omega((\log_3^n)^2)$$

$\therefore$ The time complexity is $\Theta((\log_3^n)^2)$

First time $m=1$ $2\log m = 2\log 1$

Second time $m=3$ $2\log m = 2\log 3$

Third time $m=9$ $2\log m = 2\log 9$

$\vdots$

n time $m = 3^k$ $2\log m = 2\log 3^k$

$3^k \geq n$

$\log_3 3^k \geq \log_3^n$

$k\log_3 3 \geq \log_3^n$

$k \geq \log_3^n$

$\therefore$

$$\sum_{m=1}^{\lfloor \log_3^n \rfloor} \lceil 2\log m \rceil = 2\log 1 + 2\log 3 + \cdots + 2\log 3^{\log_3^n}$$

$$= 2(\log 1 + \log 3 + \cdots + \log 3^{\log_3^n})$$

$$= 2\log(1 \times 3 \times \cdots \times 3^{\log_3^n})$$

$$= 2\log(3^0 \times 3^1 \times 3^2 \times \cdots \times 3^{\log_3^n})$$

$$= 2\log 3^{(0+1+2+\cdots+\log_3^n)} = 2\log 3^{\frac{\log_3^n \cdot \log_3^{n+1}}{2}}$$

$$= \frac{\log_3^n \cdot \log_3^{n+1}}{2} \cdot 2\log_2 3$$

$$= (\log_3^n)^2$$

So $\sum_{m=1}^{\log_3^n} 2\log m \leq \sum_{m=1}^{\log_3^n} 2\log m + 1 \leq \log_3^n \cdot 2 \cdot \log_3^n + \log_3^n \in O((\log_3^n)^2)$

由 扫描全能王 扫描创建

3、 recursion - tree method:

$$T(n) = \begin{cases} 4 & n=1 \\ 5T(\frac{n}{2}) + 2n^2 & n>1 \end{cases}$$

$T(1) = 4$

$T(n) = 5T(\frac{n}{2}) + 2n^2$

$= 5 \cdot [5T(\frac{n}{4}) + 2(\frac{n}{2})^2] + 2n^2$

$= 25T(\frac{n}{4}) + 10\frac{n^2}{4} + 2n^2 = 25T(\frac{n}{4}) + 2\cdot 5(\frac{n}{2})^2 + 2n^2$

$= 5\cdot[25T(\frac{n}{8}) + 10(\frac{n}{4})^2] + 2n^2$

$= 125T(\frac{n}{8}) + 50\frac{n^2}{8} + 2n^2$ $\quad 2\cdot 5\cdot 5$

$\vdots$ $\quad 2\cdot 5\cdot 5\cdot 5$

$= 5^k T(\frac{n}{2^k}) + 2\cdot 5^{k-1}\cdot\frac{n^2}{2^k} + 2n^2$

$\frac{n}{2^k} = 1$

$n = 2^k$

$k = \lg n$

$T(n) = 5^{\lg n} T(\frac{n}{2^{\lg n}}) + 2\cdot 5^{\lg n - 1}\cdot\frac{n^2}{2^{\lg n}} + 2n^2$

$= 5^{\lg n} T(1) + 2\cdot 5^{\lg n - 1}\cdot n + 2n^2$

$= 5^{\lg n}\cdot 4 + 2\cdot 5^{\lg n-1}\cdot n + 2n^2$

$= n^{\lg 5}\cdot 4 + 2\cdot\frac{1}{5}\cdot n^{\lg 5}\cdot n + 2n^2 \quad \in \Theta(n^{\log_2 5})$

master theorem:

$T(n) = 5T(\frac{n}{2}) + 2n^2$

$a=5 \quad b=2 \quad y=2$

$x = \log_b a = \log_2 5 \approx 2.3219 > y$

$\because$ According master theorem $T(n) \in \Theta(n^x)$ if $y < x$

$\therefore T(n) \in \Theta(n^{\log_2 5})$

4. int Fiction (A :: array , n :: integer) {

```
if (n>1){
    B <- A[1], ..., A[n/3];
    C <- A[n/3+1], ..., A[2·n/3];
    D <- A[2·n/3+1], ..., A[n];
    C <- Perturb(C)
    cond2 <- Fiction(C, n/3)
    if (cond 2){
        cond 1 <- Fiction(B, n/3);
        cond3 <- Fiction(D, n/3);
        cond 2 <- (cond1 + cond3)/2;
    }
    return cond 2
}
else
    return 1;
}
```

for these step is split A into 3 subarray it actually go through all the element in A, so it $\Theta(n)$

$\rightarrow$ C is only $\frac{n}{3}$ for A(n) so it's $\Theta(\frac{n}{3}\log\frac{n}{3})$

Fiction(C, n/3) $\rightarrow T(\frac{n}{3})$

Fiction(B, n/3) $\rightarrow T(\frac{n}{3})$

Fiction(D, n/3) $\rightarrow T(\frac{n}{3})$

$\therefore T(n) = 3T(\frac{n}{3}) + n + \frac{n}{3}\log\frac{n}{3}$

$= 3\cdot[3T(\frac{n}{9}) + \frac{n}{3} + \frac{n}{9}\log\frac{n}{9}] + n + \frac{n}{3}\log\frac{n}{3}$

$= 3^2T(\frac{n}{3^2}) + n + \frac{n}{3}\log\frac{n}{3^2} + n + \frac{n}{3}\log\frac{n}{3}$

$= 3\cdot[9\ T(\frac{n}{27}) + \frac{n}{3} + \frac{n}{9}\log\frac{n}{27}] + n + \frac{n}{3}\log\frac{n}{3}$

$= 27\ T(\frac{n}{27}) + n + \frac{n}{3}\log\frac{n}{27} + n + \frac{n}{3}\log\frac{n}{3}$

$\vdots$

$= 3^k T(\frac{n}{3^k}) + n + \frac{n}{3}\log\frac{n}{3^k} + n + \frac{n}{3}\log\frac{n}{3}$

$\frac{n}{3^k} = 1 \Rightarrow n = 3^k \Rightarrow k = \log_3 n$

$= 3^{\log_3 n}T(1) + n + \frac{n}{3}\log(n^{\log_3 n})$

$= nT(1) + n + \frac{n}{3}\cdot\log_3 n\cdot\log n \in O(n(\log n)^2)$

由 扫描全能王 扫描创建

b. $T(n) = 3T(\frac{n}{3}) + \Theta(n + \frac{n}{3}\log\frac{n}{3})$

$a=3 \quad b=3 \quad y=1 \quad m=1$

$x = \log_b a = \log_3 3 = 1 = y$

$\because$ Master Theorem $T(n) \in \Theta(n^y \log^{m+1} n)$ if $y = x$

$\therefore$ $T(n) \in \Theta(n \log^2 n) = \Theta(n(\log n)^2)$

Because of $\Theta(n(\log n)^2) = O(n(\log n)^2)$

so best case is equal to worst case, so best case equal $\Omega(n(\log n)^2)$.