Assignment#3 Report
Team name: Yunfan Li, Haichao Zhang, Jingyuan Xu
Course: CS325-001

Pseudocode for TASK 1:

Method #1:

```
res <- INF
between(suml[n], tl, sumr[n], tr)
    for i <- (tl-1) down to 0
        for j <- 0 to (tr-1)
            if abs(suml[i] + sumr[j]) < res
                res <- abs(suml[i] + sumr[j])
                start <- i
                end <- j
```

Method #2:

```
res <- INF
between(suml[n], tl, sumr[n], tr)
    sort(suml[n])
    sort(sumr[n])
    for i <- 0 to (tl-1)
        min <- INF
        for j <- 0 to (tr-1)
            if min > abs(suml[i] + sumr[j])
                min <- abs(suml[i] + sumr[j])
                if res > min
                    res <- min
                    start <- suml[i]'s original position
                    end <- sumr[j]'s original position
```

Method #3:

```
res <- INF
between(suml[n], tl, sumr[n], tr)
    sort(suml[n])
    sort(sumr[n])
    combine suml[n] with sumr[n] to a new array A[2n]
    for i <- 1 to (2n-1)
        if A[i] and A[i-1] are not in the same original array
            if res > abs(abs(A[i]) - abs(A[i-1]))
                res <- abs(abs(A[i]) - absA[i-1])
                start <- A[i]'s original position
                end <- A[i-1]'s original position
```

Pseudocode for TASK 2:

```
res <- INF
suml[10000] <- {0}
sumr[10000] <- {0}
```

```
conquer(a[n], s, e)
if s = e
        if res > abs(a[s])
                start <- s
                end <- e
                res <- abs(a[s])
        return
else
        m <- (e-s+1) / 2
        mod <- (e-s+1) % 2
        kl <- 0
        kr <- m - 1
        inl <- 0
        inr <- 0

        init sumr[n] and suml[n] to 0

        conquer(a[n], s, s+m-1)
        if mod = 1
                conquer(a[n], e-m, e)
                for i <- e-m to e
                        sumr[kr].pos <- i
                        sumr[kr].val <- sumr[[kr-1].val + a[i]
                        inr <- inr + 1
        else
                conquer(a[n], e-m+1, e)
                for i <- e-m+1 to e
                        sumr[kr].pos <- i
                        sumr[kr].val <- sumr[kr-1].val + a[i]
                        inr <- inr +1
                        kr <- kr + 1
        for i <- s+m-1 down to s
                suml[kl].pos <- i
                suml[kl].val <- suml[kl+1].val + a[i]
                inl <- inl + 1
                kl <- kl – 1
        between(suml, inl, sumr, inr)
```

Recurrence Relation & Solve:

Method #1:

T(n) = 2T(n/2) + n^2

Method #2:

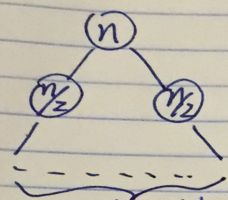
T(n) = 2T(n/2) + n^2

Method #3:

T(n) = 2T(n/2) + n

Method #1:

$$T(n) = 2T(n/2) + n^2$$



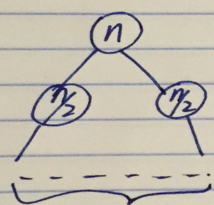
2^i problems: each $\left(\frac{n}{2^i}\right)^2$

$$\sum_{i=0}^{\log_2 n} 2^i \cdot \left(\frac{n}{2^i}\right)^2 = \sum_{i=0}^{\log_2 n} \frac{n^2}{2^i} = n^2 \cdot \sum_{i=0}^{\log_2 n} \frac{1}{2^i}$$

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

Method #2:

$$T(n) = 2T(n/2) + O(n^2)$$



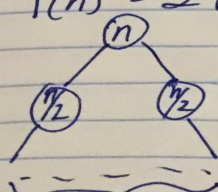
2^i problems: each $\left(\frac{n}{2^i}\right)^2$

$$\sum_{i=0}^{\log_2 n} 2^i \cdot \left(\frac{n}{2^i}\right)^2 = \sum_{i=0}^{\log_2 n} \frac{n^2}{2^i} = O(n^2) \cdot \sum_{i=0}^{\log_2 n} \frac{1}{2^i}$$

$$= O(n^2)$$

Method #3:

$$T(n) = 2T(n/2) + n$$

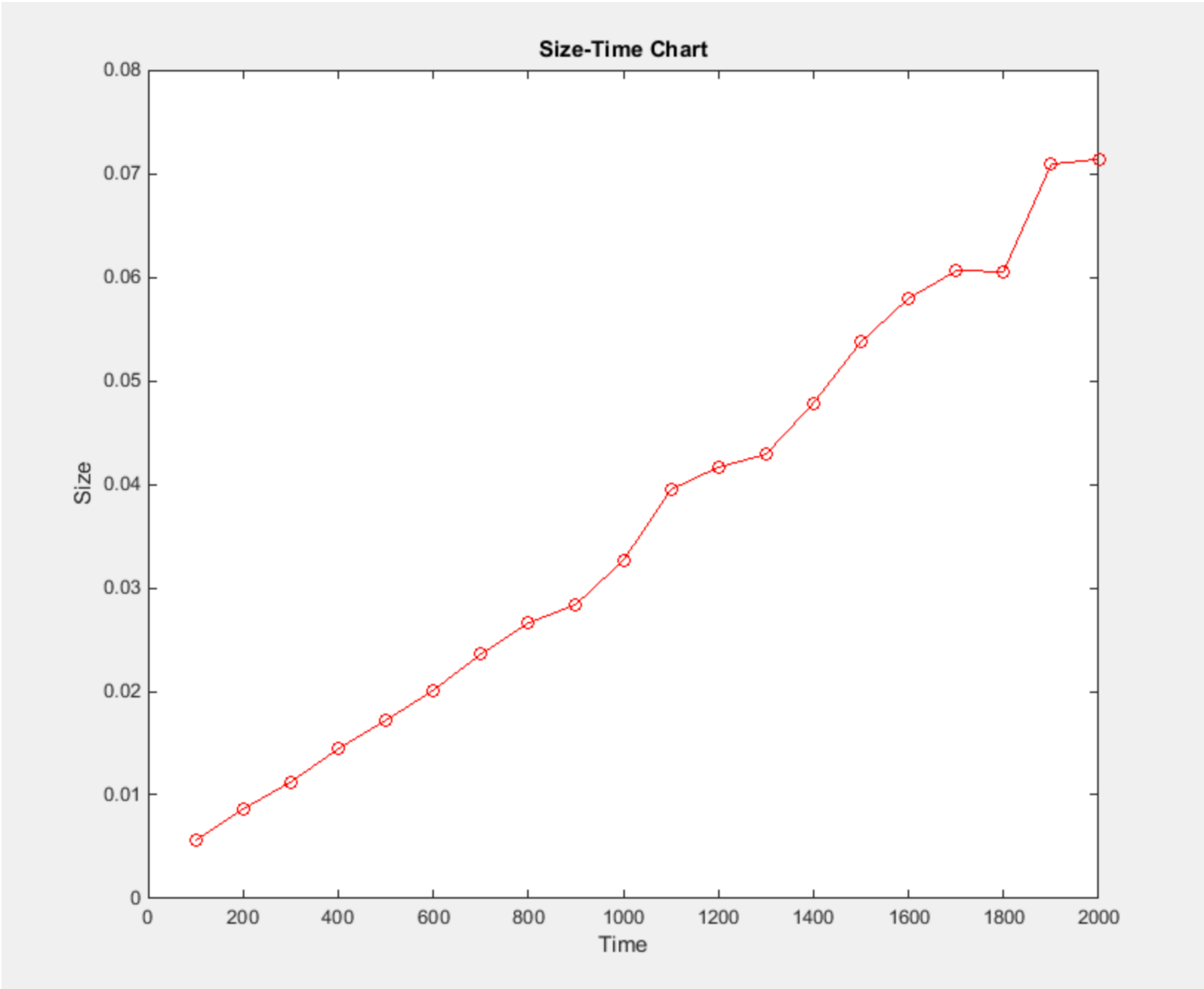


2^i problems: each $\frac{n}{2^i}$

$$\sum_{i=0}^{\log_2 n} 2^i \cdot \frac{n}{2^i} = \sum_{i=0}^{\log_2 n} n = n \cdot \sum_{i=0}^{\log_2 n} (1)$$

$$= \Theta(n \cdot \log_2 n).$$

Plot:



size	time
100	0.0056
200	0.00865
300	0.01125
400	0.01445
500	0.0172
600	0.0201
700	0.0236
800	0.0266
900	0.0284
1000	0.03265
1100	0.0395
1200	0.04165
1300	0.0429
1400	0.04785
1500	0.05375
1600	0.05795

1700	0.0606
1800	0.06055
1900	0.0709
2000	0.0714