# HW02 for ECE 9343

Tongda XU, N18100977

September 28, 2018

## 1 Question 1: 3-divide maximum subarray

MAXFROMLEFT$(A, p, r)$

```
1  max = −∞
2  for i = p to r
3        max = Sum(A, p, i) > max?Sum(A, p, i) : max
4  return max
```

MAXFROMRIGHT$(A, p, r)$

```
1  max = −∞
2  for i = r downto p
3        max = Sum(A, i, r) > max?Sum(A, i, r) : max
4  return max
```

THREE-FOLD-MAXSUB$(A, p, r)$

```
1  s = ⌊(p + r)/3⌋
2  t = ⌊(p + r)2/3⌋
3  if Sum(A, s, t − 1) > 0
4        return max(maxFromLeft(A, p, s − 1), maxFromRight(A, t, r)) + Sum(A, s, t − 1)
5  else return max(maxFromLeft(A, p, s − 1), maxFromRight(A, t, r))
```

The time complexity is $\Theta(n)$

## 2 Question 2: Intermediate Sequence

BUBBLE SORT$(A)$

$A = [11, 8, 7, 5, 3, 1]$
$\rightarrow [8, 11, 7, 5, 3, 1] \rightarrow [8, 7, 11, 5, 3, 1] \rightarrow [8, 7, 5, 11, 3, 1] \rightarrow [8, 7, 5, 3, 11, 1] \rightarrow [8, 7, 5, 3, 1, 11]$
$\rightarrow [7, 8, 5, 3, 1, 11] \rightarrow [7, 5, 8, 3, 1, 11] \rightarrow [7, 5, 3, 8, 1, 11] \rightarrow [7, 5, 3, 1, 8, 11]$
$\rightarrow [5, 7, 3, 1, 8, 11] \rightarrow [5, 3, 7, 1, 8, 11] \rightarrow [5, 3, 1, 7, 8, 11]$
$\rightarrow [3, 5, 1, 7, 8, 11] \rightarrow [3, 1, 5, 7, 8, 11]$
$\rightarrow [1, 3, 5, 7, 8, 11]$

INSERTION SORT($A$)

1 $A = [11, 8, 7, 5, 3, 1]$
2 $\rightarrow [8, 11, 7, 5, 3, 1]$
3 $\rightarrow [8, 7, 11, 5, 3, 1] \rightarrow [7, 8, 11, 5, 3, 1]$
4 $\rightarrow [7, 8, 5, 11, 3, 1] \rightarrow [7, 5, 8, 11, 3, 1] \rightarrow [5, 7, 8, 11, 3, 1]$
5 $\rightarrow [5, 7, 8, 3, 11, 1] \rightarrow [5, 7, 3, 8, 11, 1] \rightarrow [5, 3, 7, 8, 11, 1] \rightarrow [3, 5, 7, 8, 11, 1]$
6 $\rightarrow [3, 5, 7, 8, 1, 11] \rightarrow [3, 5, 7, 1, 8, 11] \rightarrow [3, 5, 1, 7, 8, 11] \rightarrow [3, 1, 5, 7, 8, 11] \rightarrow [1, 3, 5, 7, 8, 11]$

# 3 Question 3: Illustrate Merge Sort

MERGE SORT($A$)

1 $15, 16, 25, 29, 30, 40, 48$
2 $15, 29, 48 || 16, 25, 30, 40$
3 $29 || 15, 48 || 25, 40 || 16, 30$
4 $- || 48 || 15 || 40 || 25 || 16 || 30$

# 4 Question 4: CLRS Problem 2-1

## 4.1 a. show time complexity

$\Theta(T) = \frac{n}{k}\Theta(n^2) = \Theta(nk)$

## 4.2 b. show merge, c. show whole

There should not be anything special about Merge function, just use the original interface and implement of Merge in CLRS pp 31.

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

Regarding the iterative tree, it is easy to notice that: For branch (Merge), the complexity is $\Theta(nlg\frac{n}{k})$, For leaf (Insertion sort), is $\Theta(nk)$

MERGE-SORT($A, p, r, k$)

```
1  if r − p + 1 ≤ k
2        Insertion-Sort(A, p, r)
3        return
4  elseif p < r
5        q = ⌊(p + r)/2⌋
6        Merge-Sort(A, p, q)
7        Merge-Sort(A, q + 1, r)
8        Merge (A, p, q, r)
9        return
10 else return
```

# 5 Question 5: CLRS Problem 6.1-3

# 6 Question 6: CLRS Problem 6.2-6

1. Note that the height of a Heap is no more than $lg(n + \frac{1}{2}n - 1)$ in worst condition
2. Note that each round of $MAX - HEAPIFY$ takes constant time
4. Each time $MAX - HEAPIFY$ happen, the height of pointer $\leftarrow$ pointer- 1
5. We have:

$$T(h) = \begin{cases} c & h = 0 \\ T(h-1) + c & n > 0 \end{cases}$$

Solves: $T(h) = \Theta(h) = \Omega(lg\frac{3}{2}n - 1) = \Omega(lgn)$