CSCI 430: Homework 6

John E. Buckley III

October 22, 2017

1 4.1-1

FIND-MAX-SUBARRAY is supposed to find the subarray with the maximum sum, however if all elements in A are negative then FIND-MAX-SUBARRAY will return the max single negative number.

2 4.1-2

```
Maximum-Subarray(A, low, high)
    left=0
    right=0
    sum=-∞
    for i=low to high
        newsum=0
        for j=i to high
            newsum+=A[j]
            if newsum¿sum
                sum=newsum
                left=i
                right=j
return(left, right, sum)
```

3 4.1-4

If the sum of the maximum subarray is negative, then return the empty subarray.

4 4.3-1

Let us assume $T(n) \leq cn^2$ for some c. we get:
$T(n) \leq c(n-1)^2 + n = cn^2 - 2cn + c + n$

and if we pick c=1, we get:
$n^2 - 2n + 1 + n = n^2 - n + 1 \leq n^2$ for $n \geq 1$
Hence, $T(n) = T(n-1) + n$ is $O(n^2)$.

5 4.3-2

Let us assume $T(n) \leq clg(n-1)$ we get:
$T(n) \leq clg([n/2] - a) + 1$
$\leq clg([\frac{n+1}{2}] - a) + 1$
$= clg(n + 1 - 2a) - c + 1$
$\leq clg(n - a) - c + 1$ for $(a \geq \frac{1}{3})$
$\leq clg(n - a)$ for $(c \geq 1)$
Hence, $T(n) = T([n/2]) + 1isO(lgn)$.

6 4.3-3

Lets suppose $T(n) \geq cnlgn$
$T(n) \geq 2c(\frac{n}{2})lg(\frac{n}{2}) + n$
$= cnlgn - cn + n$
$\geq cnlgn$ for $(c \leq 1)$
Hence this recurrence is also $\Omega(nlgn)$ and if the upper and lower bounds are both $nlgn$ then the "exact" bound is also nlgn ($\Theta(nlgn)$).

7 4.5-1

$n = n^{\frac{1}{2}}$ because a=2 and b=4 for all occurrences.

1. $f(n) = O(1) = O(n^{\frac{1}{2}-\frac{1}{2}})$ which is case 1 of the master method hence, $T(n) = \Theta(n^{\frac{1}{2}})$
2. $f(n) = O(n^{\frac{1}{2}})$ which is case 2 of the master method hence, $T(n) = \Theta(n^{\frac{1}{2}}lgn)$
3. $f(n) = O(n) = O(n^{\frac{1}{2}+\frac{1}{2}})$ which is case 3 of the master method hence, $T(n) = \Theta(n)$
4. $f(n) = O(n^2) = O(n^{\frac{1}{2}+\frac{3}{2}})$ which is also case 3 of the master method hence, $T(n) = \Theta(n^2)$

8 4.5-2

The running time for Strassen's algorithm is $\Theta(n^{lg7})$ Professor Caesar's running time for his algorithm, in the worst case, is: $T(n) = \Theta(n^{lg_b a}) = \Theta(n^{lg_4 a}) = \Theta(n^{lg_2 \sqrt{a}})$.

For the professors algorithm to be smaller than Strassens then $n^{lg\sqrt{a}}$ must be smaller than n^{lg7}:
$n^{lg\sqrt{a}} < n^{lg7}$
$lg\sqrt{a} < lg7$
$\sqrt{a} < 7$
$a < 49$ The largest integer value of a is 48
Hence, the professor's algorithm is faster than Strassens when a¡48.

9 4.5-3

In the given recurrence a=1 and b=2. Hence, $n^{lg_2 1} = 1 \rightarrow T(n) = lgn$ Thus, $T(n) = T(n/2) + \Theta(1)$ is $T(n) = \Theta(lgn)$

10 Non-Chapter

see notes