4.1-5

Question

Use the following ideas to develop a non-recursive, linear time algorithm for the max-subarray problem. Start at the left end of the array, and progress toward the right, keeping track of the maximum subarray seen so far. Knowing a maximum subarray of $A[1..j]$, extend the answer to find a maximum subarray for index $j+1$ by using the following observation: a maximum subarray of $A[1..j+1]$ is either a maximum subarray of $A[1..j]$ or a subarray $A[i..j+1]$, for some $1 \leq i \leq j+1$. Determine a maximum subarray of the form $A[i..j+1]$ in constant time based on knowing a maximum subarray ending at index j.

Criteria

0 pts Missing, copied verbatim online, etc.

1-2 pts Does not actually seem to solve the problem

3-4 pts Solves the problem in $\omega(n)$

5 pts Good work!

Use your best judgment to fairly assess aspects of the assignment I have not includes. You may take off points for extremely sloppy or careless work.

Answer

Pseudocode : Python
(Note : Assuming the input array is a mix of atleast one positive and negative number)

```
def max_subarray(A):
    for i in range(0 , len(A)):
        max_sum += A[i]
        end += 1
        if(max_sum > sum):
            sum = max_sum
            high = end
            low = start
        elif(max_sum < 0):
            max_sum = 0
            start = high = i+1
    high = high
    low= low + 1
    return(low,high)
```

4.4-3

Question

Use a recursion tree to determine a good asymptotic upper bound on the recurrence $T(n) = 4T(n/2 + 2) + n$. Use the substitution method to verify your answer.

Criteria

0 pts Missing, copied verbatim online, etc.

1-2 pts The solution to the recurrence is incorrect (no need to look at the proof)

3-4 pts The solution is correct but the proof is wrong

5 pts Good work!

Take at least a point off if they do not state the inductive hypothesis and clearly indicate where it is used. If their solution is significantly longer than necessary or convoluted then it is conceivable that a point would be deducted.

Answer

We have $T(n) = 4T(n/2 + 2) + n$
The recursion tree can be made with four branches at level 1 of $n/2 + 2$.
The next level have 16 branches coming from its immediate parent of $n/4+1+2$ and so on.
Therefore , the height of tree will be lgn and number of leaves will be 4^i , where $i = logn$. i.e No of leaves $= 4^{logn} = n^2$
We have ,

$$T(n) = \sum_{i=0}^{\log n - 1} (2^i n + 2^{i+2}(2^i - 1)) + \theta(n^2)$$

$$= \sum_{i=0}^{\log n - 1} 2^i n + \sum_{i=0}^{\log n - 1} (2^{2i+2}) - \sum_{i=0}^{\log n - 1} (2^{i+2}) + \theta(n^2)$$

$$= \frac{2^{\log n} - 1}{2 - 1} n + 4 \sum_{i=0}^{\log n - 1} (2^{2i}) - 4 \sum_{i=0}^{\log n - 1} (2^i) + \theta(n^2)$$

$$= n^2 - 1 + 4(\frac{2^{2\log n} - 2}{4 - 1}) - 4(\frac{2^{\log n} - 1}{2 - 1}) + \theta(n^2)$$

$$= n^2 + 4/3n^2 + C_1 n + \theta(n^2)$$

On neglecting the smaller terms

$$T(n) = \theta(n^2)$$

To verify the answer , we use substitution method
let the solution be ,
$T(n) \leq cn^2 + n$

$$\begin{aligned} T(n) &\leq 4c(\frac{n}{2})^2 + \frac{n}{2} + n \\ &\leq cn^2 + \frac{3n}{2} \\ &= \theta(n^2) \end{aligned}$$

4-1

Question

Give assymptotic lower and upper bounbdsfor $T(n)$ in each of the following recurrences. Assume that $T(n)$ is constant for $n \leq 2$. Make your bounds as tight as possible, and justify your answers.

Criteria

0 pts Missing, copied verbatim online, etc.

1-4 pts Some or all of the answers were wrong (-1 per incorrect solution)

5 pts Good work!

Take of one point per incorrect answer, up to 4 points max. An answer is incorrect if the solution to the recurrence is wrong OR the rationale is missing or wrong.

Answers

(a) $T(n) = 2T(n/2) + n^4$ $T(n) = \theta(n^4)$

Using Master's Theorem , We have a=2 , b=2 and $f(n) = n^4$

$$n^{\log_b a} = n^{\log_2 2} = n$$
$$f(n) = \Omega(n^{\log_2 2+\epsilon}), \quad where \quad \epsilon = 3$$

According to case 3 of master's theorem , we have for sufficiently large n ,

$$\begin{aligned} af(\frac{n}{b}) = 2(\frac{n}{2})^4 &\leq cn^4 \\ = n^4/8 &\leq cn^4 \quad for \quad c = 1/6 \end{aligned}$$

Therefore , By Master's theorem
$T(n) = \theta(n^4)$

(b) $T(n) = T(7n/10) + n$ $\qquad T(n) = \theta(n)$

Using Master's Theorem , We have a=1 , b=10/7 and $f(n) = n$

$$n^{\log_b a} = n^{\log_{\frac{10}{7}} 1} = n^0$$
$$f(n) = \Omega(n^{\log_{\frac{10}{7}} 1+\epsilon}), \quad where \quad \epsilon = 1$$

According to case 3 of master's theorem , we have for sufficiently large n ,

$$af(\frac{n}{b}) = 7(\frac{n}{10}) \leq cn \quad for \quad c = 8/10$$

Therefore , By Master's theorem
$T(n) = \theta(n)$

(c) $T(n) = 16T(n/4) + n^2$ $\qquad T(n) = \theta(n^2 \log n)$

Using Master's Theorem , We have a=16 , b=4 and $f(n) = n^2$

$$n^{\log_b a} = n^{\log_4 16} = n^2$$

So , $f(n) = \theta(n^{log_b a})$

According to case 2 of master's theorem ,

$T(n) = \theta(n^2 \log n)$

(d) $T(n) = 7T(n/3) + n^2$ $\quad$ $T(n) = \theta(n^2)$

Using Master's Theorem , We have a=7 , b=3 and $f(n) = n^2$

$$n^{\log_b a} = n^{\log_3 7} = n^{1.77}$$
$$f(n) = \Omega(n^{\log_3 7+\epsilon}), \quad where \quad \epsilon = 0.23$$

According to case 3 of master's theorem , we have for sufficiently large n ,

$$af(\frac{n}{b}) = 7(\frac{n}{3})^2 \;\; \leq \;\; cn^2$$
$$= \frac{7}{9}n^2 \quad \leq \;\; cn^2 \quad for \quad c = 8/9$$

Therefore , By Master's theorem
$T(n) = \theta(n^2)$

(e) $T(n) = 7T(n/2) + n^2$ $\quad$ $T(n) = \theta(n^{\log 7})$

Using Master's Theorem , We have a=7 , b=2 and $f(n) = n^2$

$$n^{\log_b a} = n^{\log_2 7} = n^{2.8}$$
$$f(n) = O(n^{\log_2 7-\epsilon}), \quad where \quad \epsilon = 0.8$$

According to case 1 of master's theorem ,
$T(n) = \theta(n^{\log 7})$

(f) $T(n) = 2T(n/4) + \sqrt{n}$ $\quad$ $T(n) = \theta(\sqrt{n}\log n)$

Using Master's Theorem , We have a=2 , b=4 and $f(n) = \sqrt{n}$

$$n^{\log_b a} = n^{\log_4 2} = \sqrt{n}$$

So , $f(n) = \theta(n^{log_b a})$

According to case 2 of master's theorem ,
$T(n) = \theta(\sqrt{n}\log n)$

(g) $T(n) = T(n-2) + n^2$ $\quad\quad T(n) = \theta(n^3)$

$$\begin{aligned} T(n) &= n^2 + T(n-2) \\ &= n^2 + (n-2)^2 + T(n-4) \\ &= \sum_{i=0}^{\frac{n}{2}} (n-2i)^2 = \sum_{i=0}^{\frac{n}{2}} (n^2 - 4ni + 4i^2) \\ &= \sum_{i=0}^{\frac{n}{2}} n^2 - 4n \sum_{i=0}^{\frac{n}{2}} i + 4 \sum_{i=0}^{\frac{n}{2}} i^2 \end{aligned}$$

$$Therefore, \quad\quad T(n) = \theta(n^3)$$

4-3 (More recurrence examples)

Question

Give asymptotic lower and upper bounbdsfor $T(n)$ in each of the following recurrences. Assume that $T(n)$ is constant for $n \leq 2$. Make your bounds as tight as possible, and justify your answers.

NOTE: Most of these should not be taking youy very long - if you understand the method. There are only a few examples that need more attention.

Criteria

0 pts Missing, copied verbatim online, etc.

1-4 pts Some or all of the answers were wrong (-1 per incorrect solution)

5 pts Good work!

Take of one point per additional incorrect answer, up to 4 points max. An answer is incorrect if the solution to the recurrence is wrong OR the rationale is missing or wrong.

Answers

(a) $T(n) = 4T(n/3) + n \lg n$ $\quad\quad T(n) = \theta(n^{\log_3 4})$

Using Master's Theorem , We have a=4 , b=3 and $f(n) = n \log n$

$$n^{\log_b a} = n^{\log_3 4} = n^{1.26}$$

$$f(n) = O(n^{\log_3 4 - \epsilon}), \quad where \quad \epsilon = 0.26$$

According to case 1 of master's theorem ,
$T(n) = \theta(n^{\log_3 4})$

(b) $T(n) = 3T(n/3) + n/\lg n$ $\qquad T(n) = \theta(n \log\log n)$

We have $T(n) = 3T(n/3) + n/\log n$
The recursion tree can be made with three branches at level 1 of $\frac{n/3}{\log n/3}$.
The next level have 9 branches coming from its immediate parent of $\frac{n/9}{\log n/9}$ and so on.
Therefore , the height of tree will be $\log_3 n$ and number of leaves will be 3^i , where $i = \log_3 n$.
i.e No of leaves $= 3^{log_3 n} = n$

Cost at depth i $= 3^i * \frac{\frac{n}{3^i}}{\log n - i} = \frac{n}{\log n - i}$
We have ,

$$T(n) = \sum_{i=0}^{\log n} \frac{n}{\log n - i}$$

$$= \sum_{i=0}^{\log n} n \frac{1}{\log n - i}$$

$$= n \sum_{i=1}^{\log n} \frac{1}{\log n}$$

We know that ,

$$\sum_{i=1}^{m} \frac{1}{k} = \theta(\log m) \qquad , where \quad m -> \infty$$

Hence,

$$T(n) = \theta(n \log\log n)$$

(c) $T(n) = 4T(n/2) + n^2\sqrt{n}$ $\qquad T(n) = \theta(n^2\sqrt{n})$

Using Master's Theorem , We have a=4 , b=2 and $f(n) = n^2\sqrt{n}$

$$n^{\log_b a} = n^{\log_2 4} = n^2$$

$$f(n) = \Omega(n^{\log_2 4+\epsilon}), \quad where \quad \epsilon = 0.5$$

According to case 3 of master's theorem , we have for sufficiently large n ,

$$af(\frac{n}{b}) = 4(\frac{n}{2})^2\sqrt{(n/2)} \;\leq\; cn^2\sqrt{n}$$
$$= \frac{1}{\sqrt{2}}n^2\sqrt{n} \quad \leq \; cn^2\sqrt{n} \quad for \quad c = \frac{1}{\sqrt{2}}$$

Therefore , By Master's theorem
$T(n) = \theta(n^2\sqrt{n})$

(d) $T(n) = 3T(n/3 - 2) + n/2$ $\qquad T(n) = \theta(n \log n)$

For Sufficiently large values of n , we can ignore -2 ,and then apply master's theorem.
$T(n) = 3T(n/3) + n/2$
We have a=3 , b=3 and $f(n) = n/2$

$$n^{\log_b a} = n^{\log_3 3} = n$$

So , $f(n) = \theta(n^{log_b a})$

According to case 2 of master's theorem ,

$T(n) = \theta(n \log n)$

(e) $T(n) = 2T(n/2) + n/\lg n$ $\qquad T(n) = \theta(n \log \log n)$

We have $T(n) = 2T(n/2) + n/\log n$
The recursion tree can be made with two branches at level 1 of $\frac{n/2}{\log n/2}$.
The next level have 4 branches coming from its immediate parent of $\frac{n/4}{\log n/4}$ and so on.
Therefore , the height of tree will be $\log n$ and number of leaves will be 2^i , where $i = \log n$.
i.e No of leaves = $2^{logn} = n$

Cost at depth i = $2^i * \frac{\frac{n}{2^i}}{\log n - i} = \frac{n}{\log n - i}$
We have ,

$$T(n) = \sum_{i=0}^{\log n} \frac{n}{\log n - i}$$

$$= \sum_{i=0}^{\log n} n \frac{1}{\log n - i}$$

$$= n \sum_{i=1}^{\log n} \frac{1}{\log n}$$

We know that ,

$$\sum_{i=1}^{n} \frac{1}{k} = \theta(\log n) \qquad , where \quad n -> \infty$$

Hence,

$$T(n) = \theta(n \log \log n)$$

(f) $T(n) = T(n/2) + T(n/4) + T(n/8) + n$ $\qquad$ $T(n) = \theta(n)$

We have $T(n) = T(n/2) + T(n/4) + T(n/8) + n$
The recursion tree can be made with three branches at level 1 of $\frac{n}{2}$, $\frac{n}{4}$, $\frac{n}{8}$.
The next level have 3 more branches coming from its immediate parent of $(\frac{n}{4}$, $\frac{n}{8}$, $\frac{n}{16})$, $(\frac{n}{8}$, $\frac{n}{16}$, $\frac{n}{32})$, $(\frac{n}{16}$, $\frac{n}{32}$, $\frac{n}{64})$ and so on..
Therefore , the height of tree will be $\log n$ and number of leaves will be 3^i , where $i = \log n$.

Cost at depth i = $(\frac{7}{8})^i * n$ We have ,

$$T(n) = \sum_{i=0}^{\log n} (\frac{7}{8})^i n$$

This can be treated as a sum of an infinite GP. So,

$$T(n) = \sum_{i=0}^{\log n} \frac{1}{1 - \frac{7}{8}} n = 8n$$

Hence,

$$T(n) = \theta(n)$$

(g) $T(n) = T(n-1) + 1/n$ $\qquad$ $T(n) = \theta(\log n)$

T(n) can be expanded as follows :

$$T(n) = \frac{1}{n} + \frac{1}{n-1} + \frac{1}{n-3} + ... + \frac{1}{3} + \frac{1}{2} + \frac{1}{1}$$
$$T(n) = \sum_{k=1}^{n} \frac{1}{k} = \theta(logn) \ \ , where \quad n->\infty$$

Therefore , $T(n) = \theta(\log n)$

(h) $T(n) = T(n-1) + \lg n$ $\qquad T(n) = \theta(n \log n)$

T(n) can be expanded as follows :

$$T(n) = \log n + \log(n-1) + \log(n-3) + ... + \log 3 + \log 2 + \log 1$$
$$T(n) = \sum_{k=1}^{n} \log k = \log(n!)$$
$$T(n) \leq \log n^n = n \log n$$

Therefore , $T(n) = \theta(n \log n)$

(i) $T(n) = T(n-2) + 1/\lg n$ $\qquad T(n) = \theta(\log \log n)$

T(n) can be expanded as follows :

$$T(n) = \frac{1}{\log n} + \frac{1}{\log(n-2)} + \frac{1}{\log(n-4)} + ... + \frac{1}{\log 4} + \frac{1}{\log 2}$$
$$T(n) = \sum_{k=1}^{\frac{n}{2}} \frac{1}{\log(2k)}$$
$$T(n) = \sum_{k=1}^{\infty} \frac{1}{\log(k)}$$

Therefore , $T(n) = \theta(\log \log n)$

(j) $T(n) = \sqrt{n}T(\sqrt{n}) + n$ $\qquad T(n) = \theta(n \log \log n)$

We substitute the value of $n = m^{2^k}$,

$$T(m^{2^k}) = m^{2^{k-1}} T(m^{2^{k-1}}) + m^{2^k}$$
$$S(k) = m^{2^{k-1}} S(k-1) + m^{2^k}$$
$$= 2m^{2^k} + m^{3.2^{k-2}} S(k-2)$$

$$S(k) = \sum_{i=1}^{k}(im^{2^k} + m^{(2^i-1)2^{k-i}}S(k-i))$$

When i=k ,

$$S(k) = km^{2^k} + m^{(2^k-1)}S(0) = (k+1)m^{2^k}$$

Substituting the value of $m^{2^k} = n$ and $k = \log(\log_m n)$ in the equation ,

$$S(k) = n(\log(\log_m n) + 1) = O(n \log \log n)$$

Therefore , $T(n) = \theta(n \log \log n)$