hw2

Jingtao Scott Hong jh4ctf

September 21, 2021

1

1.1

The algorithm will implement recursive function with the help function to narrow down the index of the two in the array:

Base case:

when the size of the input array is only one, return the element in the array which is the index of the two.

Recursive case:

Run the help function and if the checks which array's sum is bigger, then divides the array into half, return the recursive functions that helps decide the difference between the sum and slicing the array of input.

1.2

Because in the helper function we are recursively slicing the two arrays into half, so it eventually when the array reach to the size of one. Thus, the run-time of f(n) is T(n) = T(n/2) + f_r.

1.3

Because comparing elements in the array and concluding the max in the summation requires run-time of n. And from previous question, we can conclude the run-time of f():

T(n)=T(n/2)+$\theta(n/2)$, and $\theta(n/2) \in O(n)$

By Master Theorem we see that Case III applies since k = 0, and is compared to n. If n is slower than n^1, then we can tell the run-time of the function is n.

2

Trying to find the pattern:
T(n)=T(n1)+n
T(n)=T(n2)+(n1)+n
T(n)=T(n-3)+(n2)+(n1)+n
.....
T(n) = n+(n-1)+(n-2)+(n-3).....+2+1+T(0) From the pattern, we can use the sequence summation theorem to calculate the run-time for it:
T(n) = n(n+1)/2 = O(n^2)

3

To inductively prove: T(n) = 4T(n/3)+n$\in$ $\emptyset(n^{log3(4)})$, we first assume T(n)$\leq$ $\frac{c * n^{\log 3(n/4)})}{3}$.
So inductively, T(n) $\leq$ $\frac{4c * n^{\log 3(n/4)})}{3}$-dn (
Thus, T(n) $\leq$ $\frac{4}{3} * c * n^{\log 3(n/4)}$-dn $\leq n^{log3(4)})$

4

Applying the Master Theorem
k = $\log_4(2) = 1/2$, f(n) = n^0 =1
Since $f(n)$=$O(n^{\log 4(2)+\epsilon})$ when ϵ =1/2, Case I applies.

5

Applying the Master Theorem
k = $\log_4(2) = 1/2$, f(n) = $n^{1/2}$
Since $f(n)$=$O(n^{\log 4(2)+\epsilon})$ when ϵ =0, Case II applies. Therefore, our run-time is $n^{1/2}$*logn.

6

Applying the Master Theorem
k = $\log_4(2) = 1/2$, f(n) = n^1=n
Since$f(n)$=$O(n^{\log 4(2)+\epsilon})$ when ϵ =1/2, Case III applies.

7

Applying the Master Theorem
k = $\log_4(2) = 1/2$, f(n) = n^2
Since $f(n)$=$O(n^{\log 4(2)+\epsilon})$ when ϵ =1/2, Case III applies.