Last Name: .... **First Name:** .... **Email:** ....

CS 6505, Fall 2017, Homework 1, 9/1/17 Due 9/8/17 in class Page 1/10

Problem 1, Analysis of Algorithm (10 points)

Where n is a power or 2 and $n > 8$, how many x's does the function Mystery(n) below print?

(a) Write and solve the recurrence exactly using substitution.

(b) State the solution in $O()$ notation.

Mystery(n)
if $n > 8$ then begin
 print("x")
 Mystery$\left(\frac{n}{2}\right)$
 Mystery$\left(\frac{n}{2}\right)$
 end

Answer:

Problem 2, Analysis of Algorithm (10 points)
Where n is a power or 2 and $n \geq 1$, how many x's does the function Mystery(n) below print?
(a) Write and solve the recurrence exactly using substitution.
(b) State the solution in $O()$ notation.

Mystery(n)
print("xx")
if $n > 1$ then begin
Mystery$\left(\frac{n}{2}\right)$
Mystery$\left(\frac{n}{2}\right)$
Mystery$\left(\frac{n}{2}\right)$
end

Answer:

Problem 3, Analysis of Algorithm (10 points)
Where n is a power or 2 and $n \geq 1$, how many x's does the function Mystery(n) below print?
(a) Write and solve the recurrence exactly using substitution.
(b) State the solution in $O()$ notation.

Mystery(n)
for $i := 1$ to n^2
    print("xx")
if $n > 1$ then begin
        Mystery$\left(\frac{n}{2}\right)$
        Mystery$\left(\frac{n}{2}\right)$
        Mystery$\left(\frac{n}{2}\right)$
        end

Answer:

Problem 4, Analysis of Algorithm (10 points)
Where n is a positive integer, how many x's does the function Mystery(n) below print?
(a) Write and solve the recurrence exactly using substitution.
(b) State the solution in $O()$ notation.

```
Mystery(n)
if n = 1 then print("x")
if n > 1 then begin
        Mystery(n − 1)
        Mystery(n − 1)
        end
```

Answer:

Problem 5, Min and Max with Fewer Comparisons (10 points)
Let $a_1 \ldots a_n$ be an input array of n unsorted distint integers, where n is an even number. Let $m_{\max}$ be the maximum value of the above integers, and let $m_{\min}$ be the minimum value of the above integers. Give an algorithm that findsboth $m_{\max}$ and $m_{\min}$ using at most $\frac{3n}{2} - 2$ comparisons.
Argue correctness and running time of your solution.

Answer:

Problem 6, Sorting Faster than $O(n \log n)$ in Special Case (10 points)
Let $a_1 \dots a_n$ be an input array of n unsorted and not necessarily distint integers. Let $m_{\max}$ be the maximum value of the above integers, and let $m_{\min}$ be the minimum value of the above integers. Let $M = m_{\max} - m_{\min}$. Give an $O(n + M)$ comparison algorithm that sorts the input array.
Argue correctness and running time of your solution.

Answer:

Problem 7, Use of Pointers (10 points)
Let $a_1 \dots a_n$ be n sorted distint integers, and let τ be an additional given integer. Give an $O(n)$ comparison algorithm that decides if there exist distinct indices i and j, ie $1 \leq i < j \leq n$, such that $a_i + a_j = \tau$.
Argue correctness and running time of your solution.

Answer:

Problem 8, Searching a Tree (10 points)
You are given a complete binary tree on n nodes, where each node has a distinct value w_i, $1 \leq i \leq n$.
The input representation is as follows:
(1) Index 1 is the root of the tree.
(2) For $1 \leq i \leq \frac{n-1}{2}$, the left child of i is $2i$ and the right child of i is $2i+1$.
(3) For $2 \leq i \leq n$, the parent of i is $\lfloor \frac{n}{2} \rfloor$.
Say that k is a *local minimum* if and only if:
(1) If $k = 1$, then w_1 is smaller than both its children.
(2) If $k \geq \frac{n-1}{2}$, then w_k is smaller than its parent.
(3) If $2 \leq k \leq \frac{n-1}{2}$, then w_k is smaller than both its children, and w_k is also smaller than its parent.
Give an $O(\log n)$ comparison algorithm that finds a local minimum of the binary tree.
Justify correctness and running time.

Answer:

Problem 9, Sorting in Linear Time in Special Case (10 points)
Suppose that n is a perfect square and let $N = n + \sqrt{n}$.
You are given an array of $a_1 \dots a_N$ distinct integers, where the first n integers are sorted $a_1 < a_2 < \dots < a_n$, but the last $\sqrt{n}$ integers are not sorted. Give an $O(N)$ comparison algorithm that sorts the entire input array $a_1 \dots a_N$.

Answer:

Problem 10, Comparing Algorithmic Performance (10 points)
Suppose you are choosing between the following three algorithms:
• Algorithm A solves problems by dividing them into five subproblems of half the size, recursively solving each subproblem, and then combining solutions in linear time.
• Algorithm B solves problems of size n by recursively solving two subproblems of size $(n-1)$ and then combining the solutions in constant time.
• Algorithm C solves problems of size n by dividing them into nine subproblems of size $n/3$, recursively solving each subproblem, and then combining the solutions in $O(n^2)$ time.
Solve each recurrence by substitution, and give the running times of each of these algorithms in $O()$ notation. Which one would you choose as the asymptotically fastest?

Answer: