Fundamental Algorithms

Homework 1

Due: June 7th, 5:30 PM EST

Instructions

Please answer each **Problem** on a separate page. Submissions must be uploaded to your account on Gradescope by the due date and time above.

Note

You have to wait until the lecture on June 2nd to solve some of the problems.

Problems To Submit

Problem 1 (20 points)

Use induction to prove the following for every positive integer n:

$$1 \times 1! + 2 \times 2! + \cdots + n \times n! = (n+1)! - 1.$$

Note that your answer should follow the following format:

1. Base case: Check that the statement holds for the base case $n = 1$.
2. Induction step:
   - Assumption: Assume that the statement holds for $n = k$.
   - Conclusion: Use the assumption to show that the statement holds for $n = k + 1$.

Recall: We have $n! = 1 \times 2 \times \cdots \times n$. Example: $5! = 1 \times 2 \times 3 \times 4 \times 5 = 120$.

Problem 2 (10+10 points)

Let's learn another sorting algorithm, SELECTION_SORT! In each iteration, SELECTION_SORT finds the largest element among the remaining elements and swap it with the last remaining element.
Example:

$$\begin{aligned} A &= [5, 3, 7, 10, 4] \\ &\Rightarrow [5, 3, 7, 4, 10] \\ &\Rightarrow [5, 3, 4, 7, 10] \\ &\Rightarrow [4, 3, 5, 7, 10] \\ &\Rightarrow [3, 4, 5, 7, 10] \end{aligned}$$

Find the time complexity of SELECTION_SORT in the best and worst case scenarios. Justify your answer.

Problem 3 (20 points)

Rank the following functions in order of their asymptotic growth. That is, find an order $f_1, f_2, f_3, \ldots, f_6$, such that $f_1 = \mathcal{O}(f_2), f_2 = \mathcal{O}(f_3)$, and so on. You do NOT need to provide an explanation of your ranking.

(a) 2^n

(b) $n \log_2 n$

(c) n^2

(d) $n!$

(e) $2^{\log_2 n}$

(f) 3^n

Problem 4 (10+10 points)

Prove or disprove: For each of the following, if it is true, then provide a reasoning, otherwise provide a counterexample and justify why the counterexample works.

(a) If $f = \mathcal{O}(g)$, then $g = \mathcal{O}(f)$.

(b) If $f = \Omega(h)$ and $g = \Omega(h)$, then $f + g = \Omega(h)$.

Problem 5 (5+15 points)

Let $A[1, \ldots, n]$ be an array of n distinct numbers. If $i < j$ and $A[i] > A[j]$, then the pair (i, j) is called a **reverse pair** for A.

(a) List the five reverse pairs of the array $A = [2, 3, 8, 6, 1]$.

(b) Modify MERGE_SORT in order to give an algorithm which determines the number of reverse pairs in an array of n numbers in $\Theta(n \log n)$ run time.
Fully explain your algorithm.

Hint: Modify the MERGE function to count the number of reverse pairs while merging!

Practice Problems

You do NOT need to submit the following problems. You can work on them for further practice.

Problem 1

Consider the following recursion with $a_1 = 3$, $a_2 = 5$, and for every integer $n \geq 2$ we have:

$$a_{n+1} = 3a_n - 2a_{n-1}.$$

Use strong induction to prove that $a_n = 2^n + 1$.
Note that your answer should follow the following format:

1. Base cases: Check that the statement holds for the base cases $n = 1, 2$.
2. Induction step:
   - Assumption: Assume that the statement holds for $n = 1, \ldots, k$.
   - Conclusion: Use the assumption to show that the statement holds for $n = k + 1$.

Problem 2

Consider the following functions which both take as arguments three n-element arrays A, B, and C:

COMPARE-1(A, B, C)
    For $i = 1$ **to** n
        For $j = 1$ **to** n
            If $A[i] + C[i] \geq B[j]$ **Return** FALSE
    Return TRUE

COMPARE-2(A, B, C)
    $aux := A[1] + C[1]$
    For $i = 2$ **to** n
        If $A[i] + C[i] > aux$ **Then** $aux := A[i] + C[i]$
    For $j = 1$ **to** n
        If $aux \geq B[j]$ **Return** FALSE
    Return TRUE

(a) When do these two functions return TRUE?

(b) What is the worst-case running time for each function?

Problem 3

Prove or disprove: For each of the following, if it is true, then provide a proof, otherwise provide a counterexample and justify why the counterexample works.

(a) If $f = \mathcal{O}(g)$ and $g = \mathcal{O}(h)$ then $f = \mathcal{O}(h)$.

(b) If $f = \Omega(g)$ and $h = \Omega(g)$ then $f = \Omega(h)$.