CS208/204: Data Structures & Abstractions
Mahidol University International College

Assignment 7: Written Answers

Austin J. Maddison

July 7, 2023

Contents

Task 1: Mathematical Truth

Proof by strong induction.

Proposition
Every binary tree on n nodes where each has either zero or two children had precisely $\frac{n+1}{2}$ leaves.

Basis Step:

$$P(1) = \frac{1+1}{2} = 1$$

This base case is true because a single node can be considered to have no children thus it the single node is a leaf.

Inductive Step: (I.H) When P(j) for $1 \leq j \leq k$ then P(k+2) is also true. The constant 2 comes from the proposition that any node can only have 2 or no children thus if we were to want to add more nodes to the tree it must be by 2 nodes at a time.

To prove that P(j) is true for $1 \leq j \leq k$ we must consider the invariance caused by the const raint. The binary tree nodes can only have 2 or no children. Suppose binary tree as a state machine starting with a single node n=1. The operation for growing a tree is constrained such that to have more nodes we have to add exactly 2 nodes to a leaf. This in turn also increases the amount of leaves by 1.

Task 3: Quick Sort Recurrence

(ii)

$$g(n) = \frac{f(n)}{n+1}$$
$$f(n) = 2 + \frac{(n+1) \cdot f(n-1)}{n}$$

$$\begin{aligned} g(n) &= \frac{f(n)}{n+1} \\ &= \frac{2 + \frac{(n+1)\cdot f(n-1)}{n}}{n+1} \\ &= \frac{2}{n+1} + \frac{f(n-1)}{n} \\ &= \frac{2}{n+1} + g(n-1) \end{aligned}$$

$$g(n) = \frac{2}{n+1} + g(n-1)$$

(iii)

$$g(n) = \frac{2}{n+1} + g(n-1)$$

The recurrence expands to:

$$g(n) = \frac{2}{n+1} + \frac{2}{n} + \frac{2}{n-1} + \frac{2}{n-2} + \frac{2}{n-3} + \cdots + \frac{2}{2}$$

$$g(n) = 2(\frac{1}{n+1} + \underbrace{\frac{1}{n} + \frac{1}{n-1} + \frac{1}{n-2} + \frac{1}{n-3} + \cdots + \frac{1}{2}}_{h(n)-1})$$

We can substitute the series with the function $h(n) - 1$ harmonic numbers.

$$g(n) = 2(\frac{1}{n+1} + h(n) - 1)$$

This completes the form of g(n) as there is no longer a recurrence relation.

(iv)

Lets first write f(n) in terms of n:

$$
\begin{aligned}
f(n) &= g(n) \cdot (n+1) \\
f(n) &= 2(\frac{1}{n+1} + \underbrace{h(n)}_{ln(n)+1 \text{ from fact given.}} -1) \cdot (n+1) \\
f(n) &= 2(\frac{1}{n+1} + (ln(n)+1) - 1) \cdot (n+1) \\
f(n) &= 2(\frac{1}{n+1} + ln(n)) \cdot (n+1)
\end{aligned}
$$

We **want to show** that $f(n)$ is $O(nln(n))$. This can be solved by the following limit:

$$
\begin{aligned}
&= \lim_{n\to\infty} \frac{f(n)}{nln(n)} \\
&= \lim_{n\to\infty} \frac{2(\frac{1}{n+1} + ln(n) \cdot (n+1))}{nln(n)} \\
&= \lim_{n\to\infty} \frac{2 + 2ln(n) \cdot (n+1)}{nln(n)} \\
&= \lim_{n\to\infty} \frac{2}{nln(n)} + \lim_{n\to\infty} \frac{2ln(n) \cdot (n+1)}{nln(n)} \\
&= \lim_{n\to\infty} \frac{2}{nln(n)} + \lim_{n\to\infty} \frac{2n+2}{n} \\
&= \lim_{n\to\infty} \frac{2}{nln(n)} + \lim_{n\to\infty} 2 + \frac{2}{n} \\
&= 2
\end{aligned}
$$

Notice when n approaches positive infinity the constant 2 is the only term remaining because other terms converge to 0. This shows that the function nln(n) times some constant can over-bound the function f(n) thus proving that f(n) is O(nln(n)).