Gabe McKay
Weicheng Li
Duran Suma

ECS 122A

Homework 1

October 8, 2025

Problem 1

1. **Inductive hypothesis**: At the end of each i^{th} loop, the variable *out* became the sum of all elements of $A[1 \ldots i]$.

   Base case ($i = 0$): The induction hypothesis is trivially true.

   Induction step: Let the variable out_i be the version of the variable at the end of i^{th} iteration. By the inductive hypothesis, out_i became the sum of all the elements in $A[1 \ldots i]$. At the end of the loop $(i+1)^{\text{th}}$, the variable out_{i+1} became $out_i + A[i+1]$, which is the sum of elements in $A[1 \ldots i+1]$.

   Wrap up: When $i = n$, the inductive hypothesis implies that the returned variable *out* became the sum of all elements in A.

2. **Inductive hypothesis**: For every complete binary tree T with height h and number x, the function `search()` returns "yes" if the tree contains x, and "no" otherwise.

   Base case ($h = 0$): The induction hypothesis is trivially true.

   Induction step: Assuming the inductive hypothesis for a non-empty input T_h with height h, consider T_{h+1} with height $h+1$. The function starts with the root $T_{h+1}.val$.

   (a) T_h is not empty, so T_{h+1} is not empty — the function won't return "no" at the beginning.
   (b) If $T_{h+1}.val = x$, the function returns "yes".
   (c) If $T_{h+1}.val > x$, the function calls search$(T_{h+1}.l, x)$ where height$(T_{h+1}.l) = h$. By the inductive hypothesis, it returns "yes" iff $T_{h+1}.l$ contains x, and "no" otherwise.
   (d) If $T_{h+1}.val < x$, the function calls search$(T_{h+1}.r, x)$ where height$(T_{h+1}.r) = h$. Similarly, it returns "yes" iff $T_{h+1}.r$ contains x, and "no" otherwise.

   Wrap up: The inductive hypothesis matches exactly the statement we set out to prove.