

Entrance Exam Report

Bryan Estrada Chiang

1/10/2026

A Gringotts

The first operation my algorithm does is to convert `knuts` into `sickles`, as long as `knuts` ≥ 29 . If so, `knuts` are subtracted by 29 and `sickles` are incremented by one. When the condition is false, then my algorithm proceed to convert `sickles` to `galleons`, as long as `sickles` ≥ 17 , and the same previous logic applies in the while-loop for this conversion.

B Sorting Hat

In order to sort the students' names and their houses in alphabetically order, my algorithm implements quicksort. First, I store the `size` of the array of students' names to sort. After that, I utilize a vector of pairs, where the first element of the pair is `name` of the student (defined as a string), and the second element of the pair is a `char` that stores the first character of the house where the student belongs. Next, my algorithm sorts the vector of pairs using a quicksort approach, where it firstly sorts the vector evaluating the students' house with pivot element selected by the quicksort function, and it secondly sorts the vector evaluation the students' names with the pivot.

C The Stairs

My algorithm finds the step with the wrong height by evaluating a few base cases first. These base cases are to determining if one of the first four steps has been changed. This is does by comparing the difference of altitudes between the step 1 to 4. If none of the first two steps has been changed, then `realHeight` is initialized to the height difference between step 1 and 2. Then, `realHeight` is compared with height difference of the i -th step with the $i + 1$ -th and $i + 2$ -th step, for $3 \leq i \leq n$. If one of those height differences is not the same as `realHeight`, then the changed-step is found by comparing the other height difference.

D Black Family Tree

The maximum number of family members connected with each other in the tree is found by using breadth-first search to traverse the family tree and count the number of nodes visited, as long as the nodes weren't visited before and they are not traitors. When there's no node to visit, the `maxComponent`, which holds the maximum number of family members connected, is updated. The tree traverse ends when all non-traitor nodes are visited or when the `maxComponent` $\leq \frac{n}{2}$.