Greedy Solution

Aditya Neeraje

June 2021

1 Introduction

This folder has the required code, in Python and C++. The Python code runs on all the test-cases in approx. 30 seconds, while the C++ code takes barely a few seconds. The idea behind the code is fairly self-obvious: given an array with first element a, last element c and all other elements forming a subarray arr, val(a, arr, c)=max(a-val(arr, c), c-val(a, arr)), where val(*args) represents the value of the array formed by concatenating all the arguments from left to right in that order. For the basecase, it is obvious that the value of a subarray of size 1 is the element of the subarray. Thus, starting from subarrays of length 1, we can iteratively fill subarrays of larger sizes, until we get the value of the only subarray of length n. Note here that the function val() measures the value of the position to Player 1. The values of the position to Player 2 are in fact the negatives of the val() values.
testcases.bash contains a bash file which automatically runs the Python code on all the testcases and returns any discrepancies. If the .sh file runs without any output, the output of the Python code is correct. Same for cpp_testcases.bash
cpp_testcases.bash takes roughly 0.6 s to be completed. testcases.bash takes roughly 22 seconds to be completed.