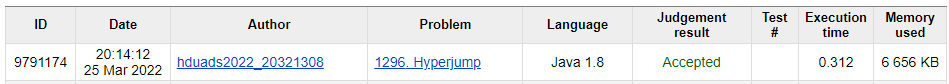
Laboratory work #1

Student: CAO Xinyang  
Student ID: 20321308  
Timus Name: hduads2022\_20321308

Mail: c.x\_yang@foxmail.com

Problem #1296

Screenshot from Timus:



Explanation of algorithm:

This code uses a dynamic programming algorithm：

First we read the input variables by scanner and store them in an array “input”. Make the initial value of "gravity\_potential" zero. Then we add “input” to “gravity\_potential” from input[0] to input[N-1]. When it's greater than 0, we think the subsequence is useful, so we keep adding. When it's less than 0, we consider the subsequence to be useless and make "gravity\_potential" zero. This means that we are going to discard this subsequence and add up the new subsequence starting with the next number. The variable “Max” is used to store the maximum value of all subsequences. When the subsequence "gravity\_potential" is greater than “Max”, then make “Max” equal to "gravity\_potential". And then we're going to get the maximum that we want.

Computational complexity of algorithm:

The for loop executes N times in this code. The complexity of the code increases linearly as N increases. So the Computational complexity of algorithm is O(n).

Source code:

import java.util.Scanner;

public class App {

public static void main(String[] args) throws Exception {

Scanner scan = new Scanner(System.in);

int N = scan.nextInt();

int Max = 0;

int gravity\_potential = 0;

int[] input = new int[N];

for(int i = 0; i < N ; i++){

input[i] = scan.nextInt();

}

for(int i = 0; i < N ; i++){

gravity\_potential += input[i];

if(gravity\_potential < 0)

gravity\_potential = 0;

if(gravity\_potential > Max)

Max = gravity\_potential;

}

System.out.println(Max);

scan.close();

}

}