ĐẠI HỌC BÁCH KHOA HÀ NỘI

Trường Công nghệ Thông tin & Truyền thông

A picture containing icon

Description automatically generated

IT3280

Thực hành Kiến trúc máy tính

BÁO CÁO MINI-PROJECT

HỌ VÀ TÊN: NGUYỄN THÀNH DƯƠNG

MSSV: 20194531

# 1.Đề bài

Bài 16: Given a sequence of integers as an array, determine whether it is possible to  
obtain a strictly increasing sequence by removing no more than one element from  
the array.  
Note: sequence a0, a1, ..., an is considered to be strictly increasing if a0 < a1 < ... <  
an. Sequences containing only one element are also considered to be strictly  
increasing.  
Example:  
- For sequence = [1, 3, 2, 1], the output should be  
almostIncreasingSequence(sequence) = false. There is no one element in this array  
that can be removed in order to get a strictly increasing sequence.  
- For sequence = [1, 3, 2], the output should be  
almostIncreasingSequence(sequence) = true. You can remove 3 from the array to  
get the strictly increasing sequence [1, 2]. Alternately, you can remove 2 to get the  
strictly increasing sequence [1, 3].

# 2.Source code

## a) Hướng giải quyết

- Kiểm tra dãy có bao nhiêu lỗi.

+Nếu không có lỗi return true.

+Nếu có nhiều hơn 1 lỗi return false.

- Nếu có 1 lỗi

+ Phần tử sai nằm ở vị trí đầu hoặc gần cuối return true.

+ Nếu phần tử vị trí i lớn hơn hoặc bằng phần tử vị trí i+1. Ta kiểm tra các điều sau.

+Remove i kiểm tra (i-1)<=(i+1) nếu đúng return true.

+Remove i+1 kiểm tra (i)<=(i+2) nếu đúng return true.

+Còn lại return false.

## b) Code C++

#include <bits/stdc++.h>

using namespace std;

bool check(int arr[], int n)

{

int count = 0;

int index = -1;

// Traverse the range [1, N - 1]

for (int i = 1; i < n; i++) {

if (arr[i - 1] >= arr[i]) {

count++;

index = i;

}

}

if (count > 1)

return false;

if (count == 0)

return true;

if (index == n - 1 || index == 1)

return true;

// a[index] is removed

if (arr[index - 1] < arr[index + 1])

return true;

// a[index - 1] is removed

if (index - 2 >= 0 && arr[index - 2] < arr[index])

return true;

// kiem tra mang rong khog

if(index < 0)

return true;

return false;

}

## c) Chú thích

- $s7 là độ dài của mảng.

- $s6 là số lỗi có phần tử vị trí i-1 lớn hơn phần tử ở vị trí i.

-$s5 là vị trí của phần tử trong mảng.

## d) Code

.data

A: .word 5,1,6

Aend: .word

Message: .asciiz "Ket qua:"

Address1: .asciiz " TRUE"

Address2: .asciiz " FALSE"

.text

main: la $a0,A

la $a1,Aend

addi $a1,$a1,-4

addi $s7,$s7,1 #do dai mang

addi $s6,$0,0 #count = 0

addi $s5,$0,-1 #index = -1

sort: beq $a0,$a1,end

addi $t0,$a0,0

demdodai:beq $t0,$a1,ret

addi $s7,$s7,1

addi $t0,$t0,4

j demdodai

ret: addi $s1, $0, 1 # i = 1

FOR: slt $t0, $s1, $s7 # $t0 = i < n?

beq $t0, $0, ENDFOR # if !(i < n) goto END

add $t1,$s1,$s1 #put 2i in $t1

add $t1,$t1,$t1 #put 4i in $t1

add $t2,$t1,$a0 #put 4i+A (address of A[i]) in $t2

lw $v0,0($t2) #gia tri i

lw $v1,-4($t2) #gia tri i-1

blt $v1,$v0,koup

addi $s6,$s6,1 #count

addi $s5,$s1,0 #index = i

koup:

addi $s1, $s1, 1 # i = i + 1

j FOR

ENDFOR: ble $s6,1,continue

j false

continue:

bnez $s6,continue2

j true

continue2:

addi $k0,$s7,-1

beq $s5,$k0,true

beq $s5,1,true

addi $s4,$s5,0

add $t1,$s4,$s4

add $t1,$t1,$t1

add $s3,$a0,$t1

lw $v0,-4($s3) #index -1

lw $v1,4($s3) #index +1

bge $v0,$v1,continue3

j true

continue3:

addi $k1,$s5,-2

bltz $k1,continue4

lw $v0,-8($s3) #index -2

lw $v1,0($s3) #index

bge $v0,$v1,continue4

j true

continue4:

bltz $s5,true

j false

true: li $v0, 59

la $a0, Message

la $a1, Address1

syscall

j end

false: li $v0, 59

la $a0, Message

la $a1, Address2

syscall

end:

## d)Kết quả đã test bằng chương trình

|  |
| --- |
|  |
|  | [1, 3, 2] returns true. |
|  | [1, 2, 1, 2] returns false. |
|  | [1, 4, 10, 4, 2] returns false; |
|  | [10, 1, 2, 3, 4, 5] returns true; |
|  | Dãy có 1 số thì end luôn . |
|  |  |