

 题目07:41:40 

P2_KIncreasing

文件上传

题目编号 1323-1514



K_Increasing



任务



使用 MIPS 汇编语言编写程序，计算使数组成为 K 递增数组所需的最少操作次数（不考虑延迟槽）。



具体要求

K 递增

给定一个下标从 0 开始包含 n 个正整数的数组 arr，和一个正整数 k。

如果对于每个满足 $k \leq i \leq n-1$ 的下标 i，都有 $arr[i-k] \leq arr[i]$ ，那么我们称 arr 是 K 递增 的。例如， $arr = [4, 1, 5, 2, 6, 2]$ 对于 $k = 2$ 是 K 递增的。

输入格式

- 第 1 行输入一个整数 n，为数组 arr 的大小 ($1 \leq n \leq 20$)
- 第 2 行至第 $n + 1$ 行，每行输入一个数字，为数组 arr 的一个元素 i
- 第 $n + 2$ 行输入一个整数 k，为 K 递增的参数值 ($1 \leq k \leq n - 1$)

输出格式

- 输出使数组成为 K 递增数组所需的最少操作次数（一次操作指修改数组中一个元素的值）

规定

- 步数限制为 200,000
- 使用 syscall 结束程序：

```
li $v0, 10  
syscall
```

题目 07:41:40

输入 ←

5
5
4
3
2
1
1

期望输出 +/-

4

样例2

输入

6
5
3
4
2
6
1
2

期望输出

3

提交要求

- 请勿使用 `.globl main`。
- 不考虑延迟槽。
- 只需要提交 `.asm` 文件。
- 程序的初始地址设置（Mars->Settings->Memory Configuration）为 `Compact, Data at Address 0`。

题目

07:41:40

```
#include <stdio.h>
#define MAX_N 20

// 计算单个子序列的 LIS 长度（最长递增子序列长度）
int computeLIS(int *subseq, int m) {
    if (m == 0) return 0;

    int dp[MAX_N];
    int maxLen = 1;

    for (int i = 0; i < m; i++) {
        dp[i] = 1;
    }

    for (int i = 1; i < m; i++) {
        for (int j = 0; j < i; j++) {
            if (subseq[j] <= subseq[i]) {
                dp[i] = dp[i] > (dp[j] + 1) ? dp[i] : (dp[j] + 1);
            }
        }
        maxLen = maxLen > dp[i] ? maxLen : dp[i];
    }

    return maxLen;
}

int kIncreasing(int* arr, int arrSize, int k){
    int ans = 0;

    for (int i = 0; i < k; i++) {
        int subseq[MAX_N];
        int m = 0;
        for (int j = i; j < arrSize; j += k) {
            subseq[m++] = arr[j];
        }
        int lisLen = computeLIS(subseq, m);
        ans += (m - lisLen);
    }
    return ans;
}

int main() {
    int n, k;
    scanf("%d", &n);
    int arr[MAX_N];
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    scanf("%d", &k);
```



提交 P2_KIncreasing



① 点击/拖拽选择文件

提交