tags: `IM2009 Algorithms` `Homework`

Algorithms Homework Assignment #10

B08705034 資管二 施芊羽

Basic Introduction

- 執行環境：Windows 10
- 使用語言：C++ 11
- Online Judger for Testing:
  - Codeforces https://codeforces.com/gym/101471/submit (https://codeforces.com/gym/101471/submit)

Problem Description

Your task is to compute the minimum achievable sum of squared errors, given parameter k and a description of the red intensities of an image's pixels. If there are n pixels that have original red values $r_1, \ldots, r_n$, and k allowed integers $v_1, \ldots, v_k$, the sum of squared errors is defined as $\sum_{i=1}^{n} min(r_i - v_i)^2$

Input and Output

Input

- $d(1 \leq d \leq 256)$: the number of distinct red values that occur in the original image
- $k(1 \leq k \leq d)$: the number of distinct red values allowed in the posterized image
- $r_1 \ldots r_d (0 \leq r \leq 255)$: a red intensity value
- $p_1 \ldots p_d (1 \leq p \leq 2^{26})$: the number of pixels having red intensity r_i

Output

The sum of the squared errors for an optimally chosen set of k allowed integer values.

Definition of Variables

- `n` : as the input d

- `k` : as the input k
- `r[]` : as the input $r_1, ... r_d$
- `p[]` : as the input $p_1, ... p_d$
- `INFINITY` : setting as `long long max` / 10 to use as ∞
- `pre_table[n + 1][n + 1]` : see the following dp description section
- `dp_table[n + 1][k + 1]` : see the following dp description section

Relationship in Dynamic Programming

`pre_table`

Denote $\text{pre\_table}[i, j]$ the minimum achievable sum of squared errors of r_i to r_j while choosing $1 \leq v \leq 255$

$\text{pre\_table}[i, j] = min(\sum_i^j (r_i - v)^2 \times p_i)$ for $v \in N$ and $0 \leq v \leq 255$

Implementation

```
long long** pre_table = new long long* [n + 1];
for(int i = 0 ; i <= n ; i++){
// initialize the pre_table
    pre_table[i] = new long long [n + 1];
    for(int j = 0 ; j <= n ; j++)
        pre_table[i][j] = INFINITY;
}
for(int l = 0 ; l < 256 ; l++){
// calculate for the minimum achievable sum of squared errors of v from 0 to 255
    for(int i = 1 ; i <= n ; i++){
    // for the i-th to j-th r
        long long ans = 0;
        for(int j = i ; j <= n ; j++){
            ans += (r[j] - l) * (r[j] - l) * p[j];
            pre_table[i][j] = min(pre_table[i][j], ans);

        }
    }
}
```

`dp_table`

Denote $\text{dp\_table}[i, j]$ the minimum achievable sum of squared errors of r_1 to r_i while choosing k allowed integer values.

$$\text{dp\_table}[i,j] = \begin{cases} 0 & \text{if } i = j = 0 \\ min(\text{dp\_table}[i,j], \text{dp\_table}[l, j-1] + \text{pre\_table}[l+1, i]) & 1 \le l \le i \end{cases}$$

Implementation

```
long long** dp_table = new long long* [n + 1];
// initialize the dp_table
for(int i = 0 ; i <= n ; i++){
    dp_table[i] = new long long[k + 1];
    for(int j = 0 ; j <= k ; j++)
        dp_table[i][j] = INFINITY;
}
dp_table[0][0] = 0; // make dp_table[0][0] to be 0
for(int i = 1; i <= n; i++){
// calculate the sum of the squared errors
    for(int j = 1 ; j <= k ; j++){
    // for an optimally chosen set of k allowed integer values
        for(int l = 0 ; l < i ; l++){
            dp_table[i][j] = min(dp_table[i][j],
                                 dp_table[l][j - 1] + pre_table[l + 1][i]);
        }
    }
}
```

Challenges I Faced

- The construction of `pre_table` and `dp_table`
  Even though I knew the concept of dynamic programming, I still found it hard to implement. I kept feeling myself stupid when making the `pre_table` and `dp_table` too small, i.e. in my previous implementation I made `dp_table` size $n \times k$. I thought I should practice more coding using the dp concept in the spring vacation.
- The setting of `INFINITY`
  In the beginning, I used `llong_max` to be the value of `INFINITY` . However, it'll cause overflow in the line of `dp_table[i][j] = min(dp_table[i][j], dp_table[l][j - 1] + pre_table[l + 1][i]);` . I think I should be more aware of this kind of situations.

Special Thanks

- My high school classmate Yi Hung who's currently studying at Dept. of CSIE in NTU provided some hints to me and stayed up late with me when I'm debugging.