hash

人员

王毅博、阮文璋、褚锦轩、王承周、许睿谦、董昱含、曹塬 到课

上周作业检查

上周作业链接: https://cppoj.kids123code.com/contest/968

2025-1008 周日13:30 (综合练习)

↻ 刷新

#	用户名	姓名	编程分	时间	A	B	C	D
1	wangchengzhou	王承周	400	6131	100	100	100	100
2	chujinxuan	褚锦轩	400	6162	100	100	100	100
3	ruanwenzhang	阮文璋	300	1694	100	100	100	
4	wangyibo	王毅博	300	4316	100	100	0	100
5	xuruiqian	许睿谦	116	27	100	16		

本周作业

https://cppoj.kids123code.com/contest/1032 (课上讲了 A ~ C 题, 课后作业是 D 题)

课堂表现

今天上课给同学们讲了 hash 这个知识点, hash 这个知识点不难, 而且可以解决很多的题目, 要求同学们必须数量掌握 hash 的写法。

课堂内容

Shortest Path 3 (上周作业)

裸 dijkstra 模板题

```cpp
#include <bits/stdc++.h>
#define int long long

using namespace std;

const int maxn = 2e5 + 5;
struct node {
  int dis, id;
  bool operator < (const node& p) const { return dis < p.dis; }
  bool operator > (const node& p) const { return dis > p.dis; }
};
vector<node> vec[maxn];
int w[maxn], f[maxn];
bool st[maxn];
```

```cpp
signed main()
{
  int n, m; cin >> n >> m;
  for (int i = 1; i <= n; ++i) cin >> w[i];
  for (int i = 1; i <= m; ++i) {
    int a, b, c; cin >> a >> b >> c;
    vec[a].push_back({c,b}), vec[b].push_back({c,a});
  }

  memset(f, 0x3f, sizeof(f));
  priority_queue<node, vector<node>, greater<node>> q;
  q.push({w[1],1}); f[1] = 0;
  while (!q.empty()) {
    node u = q.top(); q.pop();
    int dis = u.dis, id = u.id;
    if (st[id]) continue;

    st[id] = true;
    for (node it : vec[id]) {
      if (st[it.id]) continue;
      if (dis+it.dis+w[it.id] < f[it.id]) {
        f[it.id] = dis+it.dis+w[it.id]; q.push({f[it.id],it.id});
      }
    }
  }

  for (int i = 2; i <= n; ++i) cout << f[i] << " ";
  cout << endl;
  return 0;
}
```

P10468 兔子与兔子

```cpp
// 单 hash
#include <bits/stdc++.h>

using namespace std;

typedef unsigned long long ULL;
const int maxn = 1e6 + 5;
const int P = 131;
char s[maxn];
ULL p[maxn], h[maxn];

ULL get_hash(int l, int r) { return h[r] - h[l-1]*p[r-l+1]; }

int main()
{
  cin >> (s+1);
  int n = strlen(s+1);
```

```cpp
    p[0] = h[0] = 1;
    for (int i = 1; i <= n; ++i) {
        p[i] = p[i-1]*P, h[i] = h[i-1]*P + s[i];
    }

    int m; cin >> m;
    while (m -- ) {
        int l1, r1, l2, r2; cin >> l1 >> r1 >> l2 >> r2;
        if (get_hash(l1,r1) == get_hash(l2,r2)) cout << "Yes" << endl;
        else cout << "No" << endl;
    }

    return 0;
}
```

```cpp
// 双 hash
#include <bits/stdc++.h>
#define x first
#define y second

using namespace std;

typedef long long LL;
typedef pair<int, int> PII;
const int maxn = 1e6 + 5;
const int mod1 = 1e9+7, mod2 = 1e9+9;
const int P = 131;
char s[maxn];
PII p[maxn], h[maxn];

PII get_hash(int l, int r) {
    int h1 = (h[r].x - (LL)h[l-1].x*p[r-l+1].x%mod1 + mod1) % mod1;
    int h2 = (h[r].y - (LL)h[l-1].y*p[r-l+1].y%mod2 + mod2) % mod2;
    return {h1, h2};
}

int main()
{
    cin >> (s+1);
    int n = strlen(s+1);
    p[0] = h[0] = {1, 1};
    for (int i = 1; i <= n; i++) {
        h[i].x = ((LL)h[i-1].x*P + s[i]) % mod1;
        h[i].y = ((LL)h[i-1].y*P + s[i]) % mod2;
        p[i].x = (LL)p[i-1].x * P % mod1;
        p[i].y = (LL)p[i-1].y * P % mod2;
    }

    int m; cin >> m;
    while (m -- ) {
        int l1, r1, l2, r2; cin >> l1 >> r1 >> l2 >> r2;
```

```
        if (get_hash(l1,r1) == get_hash(l2,r2)) cout << "Yes" << endl;
        else cout << "No" << endl;
    }
    return 0;
}
```

[CSP-S 2023] 消消乐

csp-s 真题: 栈 + hash 维护

栈里面维护单个字符, 以及整个栈的 hash 值

```cpp
#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
typedef unsigned long long ULL;
const int maxn = 2e6 + 5;
const int P = 131;
char s[maxn];
struct node {
  char c;
  ULL val;
};

int main()
{
  int n; cin >> n >> (s+1);
  map<ULL, int> mp;
  ULL hVal = 1; ++mp[hVal];
  stack<node> stk; stk.push({' ', 1});
  for (int i = 1; i <= n; ++i) {
    if (!stk.empty() && s[i]==stk.top().c) stk.pop();
    else stk.push({s[i], stk.top().val*P+s[i]});

    ++mp[stk.top().val];
  }

  LL res = 0;
  for (auto it : mp) res += (LL)it.second*(it.second-1) / 2;
  cout << res << endl;
  return 0;
}
```

匹配统计

hash + 二分 维护

枚举点 i 作为起点, 针对 i 作为起点时, 二分长度

```cpp
#include <bits/stdc++.h>

using namespace std;

typedef unsigned long long ULL;
const int maxn = 2e5 + 5;
const int P = 131;
char a[maxn], b[maxn];
ULL p[maxn], ha[maxn], hb[maxn];
int f[maxn];

ULL get_hash(int l, int r, ULL h[]) { return h[r] - h[l-1]*p[r-l+1]; }

int calc(int id, int n, int m) {
  int l = 1, r = min(n-id+1, m);
  while (l <= r) {
    int mid = (l + r) / 2;
    if (get_hash(id,id+mid-1,ha) == get_hash(1,mid,hb)) l = mid+1;
    else r = mid-1;
  }
  return r;
}

int main()
{
  int n, m, T; cin >> n >> m >> T;
  cin >> (a+1) >> (b+1);
  p[0] = ha[0] = hb[0] = 1;
  for (int i = 1; i < maxn; ++i) p[i] = p[i-1]*P;
  for (int i = 1; i <= n; ++i) ha[i] = ha[i-1]*P + a[i];
  for (int i = 1; i <= m; ++i) hb[i] = hb[i-1]*P + b[i];

  for (int i = 1; i <= n; ++i) {
    int len = calc(i, n, m);
    ++f[len];
  }

  while (T -- ) {
    int x; cin >> x; cout << f[x] << endl;
  }
  return 0;
}
```