trie树

人员

李锦澍、徐思远、陈欣妙、刘奕辰、袁晨峻 到课

上周作业检查

上周作业链接: https://cppoj.kids123code.com/contest/1208

王向东老师周日八点半C++综合练习

刷新

#	用户名	姓名	编程分	时间	A	B	C	D
1	xusiyuan	徐思远	400	11726	100	100	100	100
2	yangjunyan	杨俊彦	400	23775	100	100	100	100
3	liuyichen	刘奕辰	392	1878	100	100	100	92
4	chenxinmiao	陈欣妙	300	3430	100	100	100	
5	suitianyi	隋天乙	300	8944	100	100	100	
6	zhouzhirun	周治润	300	10978	100	100	100	
7	lijinshu	李锦澍	300	15003	100	100	100	
8	yangyongcheng	杨咏丞	200	9226	100	100		
9	caoyuan	曹塬	100	11142	100			

本周作业

https://cppoj.kids123code.com/contest/1339 (课上讲了 A ~ C 题, 课后作业是 D 题)

课堂表现

今天课上讲了 trie 树这个知识点, 这个知识点比较抽象, 同学们课下要好好复习一下。

课堂内容

Family and Insurance (上周作业)

输入时, 先把结果存到 f 数组中

f[i]: 在第 i 个人上最多往后传多少代

最后做一遍 dfs 搜索, 记录哪些点会被保险即可

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

using namespace std;

const int maxn = 3e5 + 5;
vector<int> vec[maxn];
int f[maxn], res = 0;
```

```
void dfs(int u, int cnt) {
  cnt = max(cnt, f[u]);
  if (cnt) ++res;
  for (int i : vec[u]) dfs(i, max(0,cnt-1));
}

int main()
{
  int n, m; cin >> n >> m;
  for (int i = 2; i <= n; ++i) {
    int x; cin >> x; vec[x].push_back(i);
  }
  while (m -- ) { int a, b; cin >> a >> b; f[a] = max(f[a], b+1); }

  dfs(1, 0);

  cout << res << endl;
  return 0;
}
```

【模板】字典树

trie 树模板题

tr[i][j]: 编号为 i 的点的 j 号孩子的编号是多少

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

using namespace std;

const int N = 3e6 + 5, M = 62;
int tr[N][M], cnt[N], idx = 0;

int get_int(char x) {
  if (islower(x)) return x-'a';
  if (isupper(x)) return x-'A'+26;
  return x-'0'+52;
}

void tr_insert(string s) {
  int p = 0;
  for (char i : s) {
    int u = get_int(i);
    if (!tr[p][u]) tr[p][u] = ++idx;
    p = tr[p][u];
    ++cnt[p];
  }
}

int tr_query(string s) {
```

```
  int p = 0;
  for (char i : s) {
    int u = get_int(i);
    if (!tr[p][u]) return 0;
    p = tr[p][u];
  }
  return cnt[p];
}

void solve() {
  int n, m; cin >> n >> m;
  while (n -- ) {
    string s; cin >> s; tr_insert(s);
  }

  while (m -- ) {
    string s; cin >> s;
    cout << tr_query(s) << "\n";
  }

  for (int i = 0; i <= idx; ++i) {
    cnt[i] = 0;
    for (int j = 0; j < M; ++j) tr[i][j] = 0;
  }
  idx = 0;
}

int main()
{
  int T; cin >> T;
  while (T -- ) solve();
  return 0;
}
```

最大异或对 The XOR Largest Pair

从 1~i 遍历, trie 树中存前 i-1 个点的二进制

对于第 i 个点, 去前面找跟 a[i] 的二进制尽量相反的二进制

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

using namespace std;

const int N = 100000 + 5, M = 32;
int tr[N*M][2], idx = 0;

void tr_insert(int x) {
  int p = 0;
  for (int i = 31; i >= 0; --i) {
    int u = (x>>i)&1;
```

```
    if (!tr[p][u]) tr[p][u] = ++idx;
    p = tr[p][u];
  }
}

int tr_query(int x) {
  int p = 0, res = 0;
  for (int i = 31; i >= 0; --i) {
    int u = (x>>i)&1;
    if (tr[p][u^1]) p = tr[p][u^1], res += ((u^1)<<i);
    else p = tr[p][u], res += (u<<i);
  }
  return res;
}

int main()
{
  int n; cin >> n;
  int res = 0;
  for (int i = 1; i <= n; ++i) {
    int x; cin >> x; tr_insert(x);
    int t = tr_query(x);
    res = max(res, x^t);
  }
  cout << res << endl;
  return 0;
}
```

最长异或路径

先 dfs 求出根到每个点的异或值, 然后问题就转化为上一个问题了

点 v1 到 v2 之间的路径异或值, 就是 根到v1的异或值 ^ 根到v2的异或值

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

using namespace std;

const int maxn = 100000 + 5;
struct node {
  int to, value;
};
vector<node> vec[maxn];

int tr[maxn*32][2], idx = 0;
void tr_insert(int x) {
  int p = 0;
  for (int i = 31; i >= 0; --i) {
    int u = (x>>i)&1;
    if (!tr[p][u]) tr[p][u] = ++idx;
    p = tr[p][u];
```

```
  }
}
int tr_query(int x) {
  int p = 0, res = 0;
  for (int i = 31; i >= 0; --i) {
    int u = (x>>i)&1;
    if (tr[p][u^1]) p = tr[p][u^1], res += ((u^1)<<i);
    else p = tr[p][u], res += (u<<i);
  }
  return res;
}

int f[maxn], res = 0;
void dfs(int u, int fa, int val) {
  f[u] = val;

  tr_insert(val);
  int val2 = tr_query(val);
  res = max(res, val^val2);

  for (node it : vec[u]) {
    if (it.to != fa) dfs(it.to, u, val^it.value);
  }
}

int main()
{
  int n; cin >> n;
  for (int i = 1; i <= n-1; ++i) {
    int u, v, w; cin >> u >> v >> w;
    vec[u].push_back({v, w}), vec[v].push_back({u, w});
  }

  dfs(1, -1, 0);

  cout << res << endl;
  return 0;
}
```