代码库

上海交通大学

August 25, 2015

Contents

1 图论

1.1 有根树的同构

```
const unsigned long long MAGIC = 4423;

unsigned long long magic[N];
std::pair<unsigned long long, int> hash[N];

void solve(int root) {
        magic[0] = 1;
        for (int i = 1; i <= n; ++i) {
                magic[i] = magic[i - 1] * MAGIC;
        }
        std::vector<int> queue;
        queue.push_back(root);
        for (int head = 0; head < (int)queue.size(); ++head) {
                int x = queue[head];
                for (int i = 0; i < (int)son[x].size(); ++i) {
                        int y = son[x][i];
                        queue.push_back(y);
                }
        }
        for (int index = n - 1; index >= 0; --index) {
                int x = queue[index];
                hash[x] = std::make_pair(0, 0);

                std::vector<std::pair<unsigned long long, int> > value;
                for (int i = 0; i < (int)son[x].size(); ++i) {
                        int y = son[x][i];
                        value.push_back(hash[y]);
                }
                std::sort(value.begin(), value.end());

                hash[x].first = hash[x].first * magic[1] + 37;
                hash[x].second++;
                for (int i = 0; i < (int)value.size(); ++i) {
                        hash[x].first = hash[x].first * magic[value[i].second] + value[i].first;
                        hash[x].second += value[i].second;
                }
                hash[x].first = hash[x].first * magic[1] + 41;
                hash[x].second++;
        }
}
```