递归存储输出二叉树

人员

洪晨栋、洪晨棋、陶汇笙 到课, 罗启宸 线上

作业检查

上周作业链接: https://cppoj.kids123code.com/contest/99

2025-0621周六10:30 (递归求排列)

刷新

#	用户名	姓名	总分	选择分	编程分	时间	A	B	C	D	E
1	wangenze	王恩泽	400	0	400	1184	100	100	100		100
2	taohuisheng	陶汇笙	300	0	300	1147	100	100	100		
3	hongchenqi	洪晨棋	300	0	300	1227	100	100	100		
4	guoxurui	郭栩睿	220	0	220	117	100	100	20		
5	luoqichen	罗启宸	200	0	200	67	100	100			
6	songjixiang	宋吉相	200	0	200	69	100	100			
7	hongchendong	洪晨栋	200	0	200	69	100	100			
8	cuichenhe	崔宸赫	200	0	200	122	100	100			

作业

https://cppoj.kids123code.com/contest/129 (课上讲了 A ~ C 这些题, 课后作业是 D 题)

课堂表现

今天的 B、C 两道关于 二叉树 的题同学们课上都做的比较好, 对同学们提出表扬

A 题课上整体做的不太好, 同学们基本都是在老师指导下做出来的, 课下要重新做一遍这道题。

课堂内容

P2089 烤鸡

一共 10 项, 每项填 1/2/3 其中一个, 问: 凑出 n 有多少方案

当 n<10 或者 n>30 一定是无解的, 在 10~30 之间, 可以递归搜索, 每一项都可以尝试填 1/2/3 其中一个

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

using namespace std;

const int maxn = 10 + 5;
int a[maxn];
int n, res = 0;;
```

```
void dfs(int u, int sum, bool flag) {
    if (u == 11) {
        if (sum == n) {
            if (flag == false) res++;
            else {
                for (int i = 1; i <= 10; i++) cout << a[i] << " ";
                cout << endl;
            }
        }
        return;
    }
    for (int i = 1; i <= 3; i++) {
        a[u] = i;
        dfs(u+1, sum+i, flag);
    }
}

int main()
{
    cin >> n;
    if (n<10 || n>30) cout << 0 << endl;
    else {
        dfs(1, 0, false);
        cout << res << endl;
        dfs(1, 0, true);
    }
    return 0;
}
```

B3833 [NICA #2] 爱与不爱

把每一项变成对应的 2 的幂次方, 然后用一个 multiset 维护, 每次取最大值和最小值, 让最大值除2, 最小值乘2, 看能否让总和变小, 直到不能让总和变小为止

```
#include<bits/stdc++.h>
using namespace std;
const int maxn=1e5+20;
multiset<int>mt;
int change(int x)
{
    int ans=0;
    while(x/2>=1)
    {
        ans++;
        x/=2;
    }
    int sum=1;
    for(int i=1;i<=ans;++i){
        sum*=2;
    }
```

```
    return sum;
}
int main()
{
    int n;
    cin>>n;
    for(int i=1;i<=n;++i){
        int q;cin>>q;
        mt.insert(change(q));
    }
    while(1){
        int tmin=*mt.begin();
        int tmax=*mt.rbegin();
        if(tmin<tmax/2)
        {
            mt.erase(mt.find(tmin));
            mt.erase(mt.find(tmax));
            mt.insert(tmin*2);
            mt.insert(tmax/2);
        }
        else break;
    }
    long long cnt=0;
    for(auto i:mt){
        cnt+=i;
    }
    cout<<cnt;
    return 0;
 }
```

B3642 二叉树的遍历

二叉树的 前中后 序遍历, 可以用一个二维数组存储二叉树, 然后递归遍历

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

using namespace std;

const int maxn = 1e6 + 5;
int tr[maxn][2];

void dfs_pre(int u) {
  cout << u << " ";
  for (int i = 0; i < 2; ++i) {
    if (tr[u][i]) dfs_pre(tr[u][i]);
  }
}

void dfs_mid(int u) {
  if (tr[u][0]) dfs_mid(tr[u][0]);
  cout << u << " ";
```

```
  if (tr[u][1]) dfs_mid(tr[u][1]);
}

void dfs_suf(int u) {
  for (int i = 0; i < 2; ++i) {
    if (tr[u][i]) dfs_suf(tr[u][i]);
  }
  cout << u << " ";
}

int main()
{
  int n; cin >> n;
  for (int i = 1; i <= n; ++i) cin >> tr[i][0] >> tr[i][1];

  dfs_pre(1); cout << endl;
  dfs_mid(1); cout << endl;
  dfs_suf(1); cout << endl;
  return 0;
}
```

P4913 【深基16.例3】二叉树深度

在 dfs 遍历 二叉树的同时, 额外传一个参数代表目前点的深度, 对所有点的深度求一个最大值

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

using namespace std;

const int maxn = 1e6 + 5;
int tr[maxn][2], f[maxn];

void dfs(int u, int depth) {
  f[u] = depth;
  if (tr[u][0]) dfs(tr[u][0], depth+1);
  if (tr[u][1]) dfs(tr[u][1], depth+1);
}

int main()
{
  int n; cin >> n;
  for (int i = 1; i <= n; ++i) cin >> tr[i][0] >> tr[i][1];

  dfs(1, 1);
  int res = 0;
  for (int i = 1; i <= n; ++i) res = max(res, f[i]);
  cout << res << endl;
  return 0;
}
```