STL

- upper_bound() :返回第一个大于val的地址(迭代器)
- lower_bound()：返回第一个大于等于val的地址(迭代器)

实现代码：

- 数组:

```
int r[] = { 1,2,4,5};
lower_bound(r+0, r + 4, val)返回第一个大于等于val的地址
upper_bound(r+0, r + 4, val)返回第一个大于val的地址
cout << lower_bound(r + 0, r + 4, val) - r;//返回对应下标
```

- 容器：

```
multiset<int>s = { 1,2,2,3,4 };
auto p = upper_bound(s.begin(), s.end(), 2);
```

作用原理：**二分**查找

注意事项：需要保证**有序**

```
111
112
113

121
122
123

131
132
133
```

搜索入门

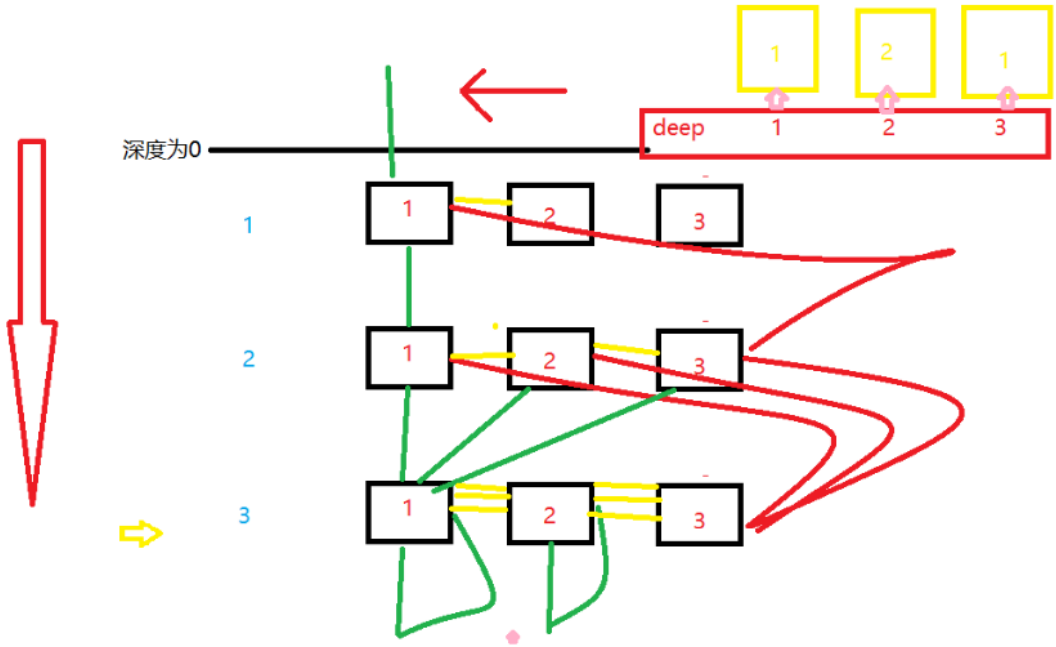
> 你面前有三个盒子，手里有三种不限量的手牌[1,2,3]，如何输出其所有可能？？

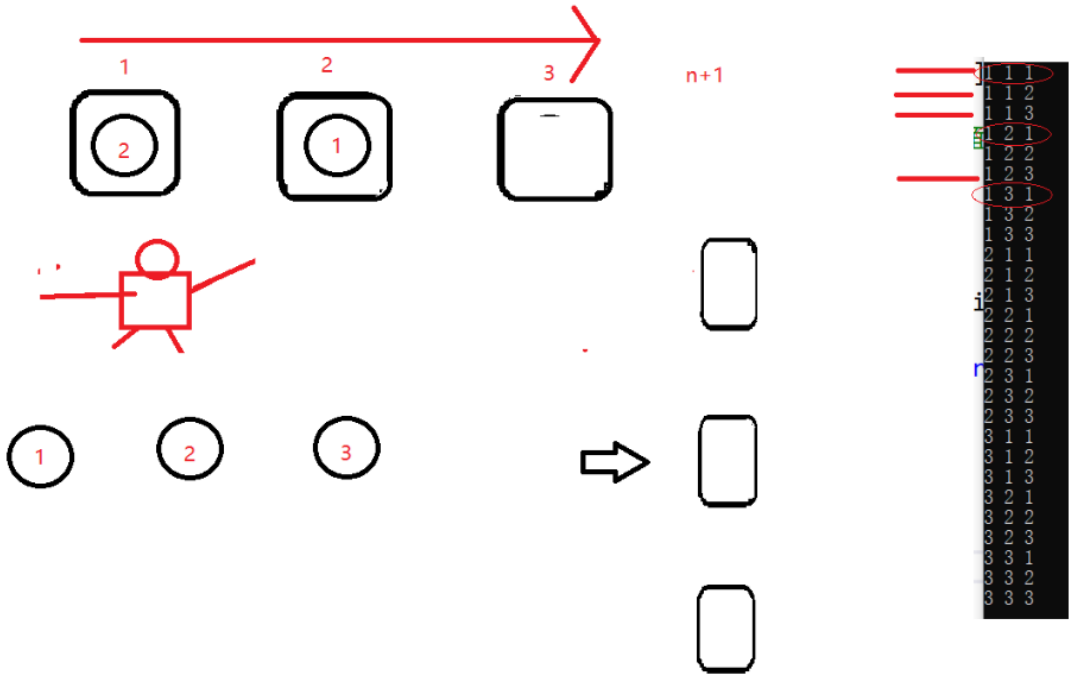
1. ```
   for (int i = 1; i <= 3; i++) {
           for (int j = 1; j <= 3; j++) {
               for (int k = 1; k <= 3; k++) {
                   cout << i << " " << j << " " << k << endl;
               }
           }
       }
   ```
   ```

带来的问题：无法自由控制n的情况

2. dfs

用一种递归实现的搜索策略






```
int r[4];
int k;
void f(int k) {
    if (k == 4) {
        for (int i = 1; i <= 3; i++) {//告诉路线
            cout << r[k] << " ";
        }
        cout << endl;
        return;
```

```
    }
    for (int i = 1; i <= 3; i++) {
        r[k] = i;
        //cout << i << " ";
        f(k + 1);
    }
}
```

3. 结果树