

# 《数据结构与算法》第六周作业反馈

助教-陈孙一硕

## A 模拟队列

直接用 STL 中的 queue 进行操作即可。

```
#include<bits/stdc++.h>
using namespace std;
#define int long long
void solve(){
    int m; cin>>m;
    queue<int> q;
    while (m--){
        string op; cin>>op;
        if (op=="push"){
            int x; cin>>x;
            q.push(x);
        }else if (op=="pop") q.pop();
        else if (op=="empty"){
            if (q.empty()) cout<<"YES"<<endl;
            else cout<<"NO"<<endl;
        }else cout<<q.front()<<endl;
    }
}
signed main(){
    ios::sync_with_stdio(false);
    cin.tie(0);
    int T=1; //cin>>T;
    while (T--) solve();
}
```

## B 字符串匹配问题 (strs)

个人认为这是一个长脑子的题目 (bushi) . 很经典的用栈进行的括号匹配问题, 码量可能稍微有点大. 我猜有很多同学第一发提交的时候应该是一排段错误 (包括我) . 需要注意的是, 对于栈和队列这样的数据结构, 每次调用栈顶或者对头元素, 以及出栈和出队操作时, 都需要对栈和队列进行判空. 否则就会出现一堆段错误的情况.

```
#include<iostream>
#include<stack>
using namespace std;
#define int long long
stack<char> stk;
bool solve(string s){
    while (!stk.empty()){
        stk.pop();
    }
    for (size_t i=0;i<s.size();i++){
        if (s[i]=='<'){
            stk.push(s[i]);
        }else if (s[i]=='>'){
            if (!stk.empty()&&stk.top()=='<'){
                stk.pop();
            }else{
                return 0;
            }
        }else if (s[i]=='('){
            if (!stk.empty()&&stk.top()=='<'){
                return 0;
            }else{
                stk.push(s[i]);
            }
        }else if (s[i]==')'){
            if (!stk.empty()&&stk.top()=='('){
                stk.pop();
            }else{
                return 0;
            }
        }
    }
}
```

```

    }else if (s[i]=='['){
        if (!stk.empty() && (stk.top()=='<' || stk.top()=='(')){
            return 0;
        }else{
            stk.push(s[i]);
        }
    }else if (s[i]==']'){
        if (!stk.empty() && stk.top()=='['){
            stk.pop();
        }else{
            return 0;
        }
    }else if (s[i]=='{'){
        if (!stk.empty() && (stk.top()=='<' || stk.top()=='(' || stk.top()=='[')){
            return 0;
        }else{
            stk.push(s[i]);
        }
    }else if (s[i]=='}'){
        if (!stk.empty() && stk.top()=='{'){
            stk.pop();
        }else{
            return 0;
        }
    }
}

if (!stk.empty()){
    return 0;
}

return 1;
}

signed main(){
    int n;
    cin>>n;
    string s;
    getchar();

```

```

while (n--){
    getline(cin,s);
    if (solve(s)){
        cout<<"YES"<<endl;
    }else{
        cout<<"NO"<<endl;
    }
}
return 0;
}

```

## C 表达式转换

模版题。用栈进行操作。

```

#include<bits/stdc++.h>
using namespace std;
int main() {
    string s;
    cin >> s;
    stack<char> st;
    int priority[128] = {0};
    priority['+'] = priority['-'] = 1;
    priority['*'] = priority['/'] = 2;
    priority['('] = priority[')'] = 3;
    bool first = true;
    for (int i = 0; i < s.length(); i++) {
        char c = s[i];
        if (((i == 0 || s[i-1] == '(') && (c == '+' || c == '-')) ||
            isdigit(c) || c == '.') {
            if (first) first = false;
            else cout << " ";
            if (c != '+') cout << c;
            while (i + 1 < s.length() && (s[i+1] == '.' || isdigit(s[i+1]))) {
                i++;
                cout << s[i];
            }
        }
    }
}

```

```

    }
}
else {
    if (c == ')') {
        while (!st.empty() && st.top() != '(') {
            cout << " " << st.top();
            st.pop();
        }
        st.pop();
    }
    else if (st.empty() || priority[c] > priority[st.top()]) {
        st.push(c);
    }
    else {
        while (!st.empty() && st.top() != '(') {
            cout << " " << st.top();
            st.pop();
        }
        st.push(c);
    }
}
}
while (!st.empty()) {
    cout << " " << st.top();
    st.pop();
}
return 0;
}

```

## D 直接插入排序

模拟插入排序的操作即可。

```

#include<bits/stdc++.h>
using namespace std;
#define int long long

```

```

const int N=2e6+101;
int n;
int a[N];
void solve(int n){
    vector<int> a(n+1);
    for (int i=1;i<=n;i++) cin>>a[i];
    for (int i=2;i<=n;i++){
        for (int j=1;j<i;j++){
            if (a[j]>a[i]){
                int x=a[i];
                for (int k=i-1;k>=j;k--){
                    a[k+1]=a[k];
                }
                a[j]=x;
                break;
            }
        }
        for (int i=1;i<n;i++) cout<<a[i]<<" ";
        cout<<a[n]<<endl;
    }
}
signed main(){
    ios::sync_with_stdio(false);
    cin.tie(0);
    while (cin>>n) solve(n);
}

```