

栈的基本概念

栈的简介

栈 (stack) 是以一种线性存储结构，它具有一下特点：栈中的数据元素遵守“先进后出 (First in Last out)”的原则，简称FILO结构；限定只能在栈顶进行插入和删除操作。

栈的相关概念

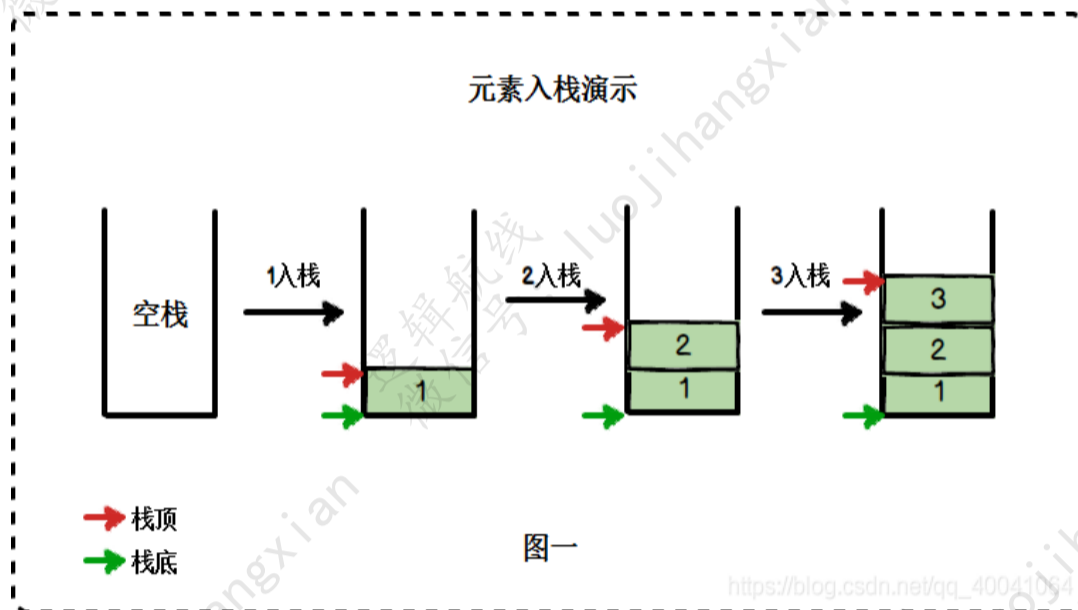
栈顶与栈底：允许元素插入与删除的一端称为栈顶，另一端成为栈底。

压栈：栈的插入操作，叫做进栈，也称压栈，入栈。

弹栈：栈的删除操作，也叫做出栈。

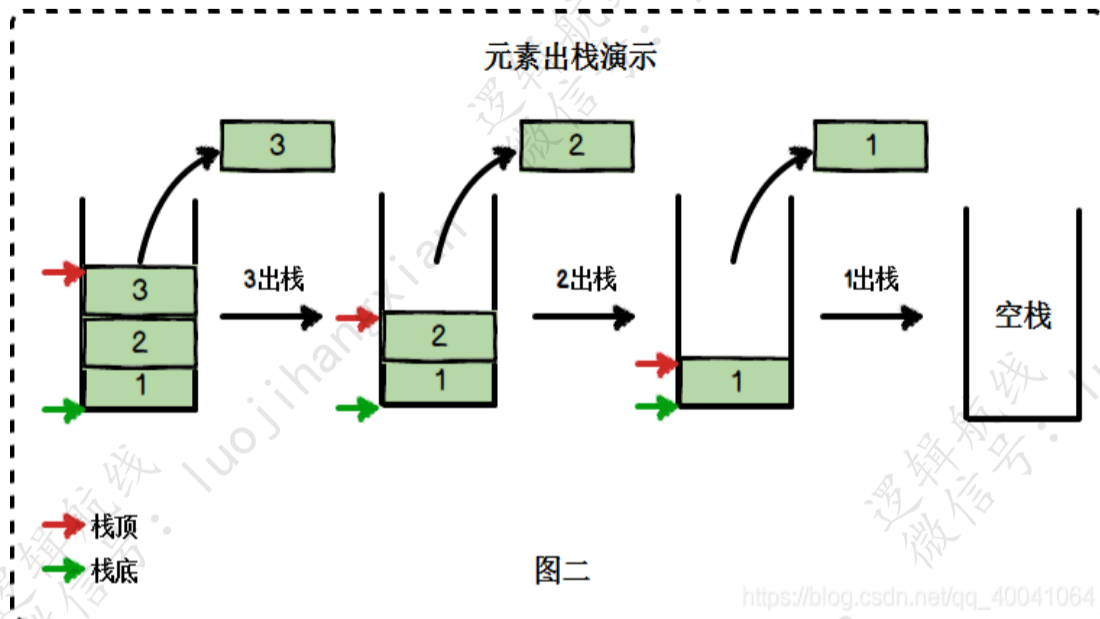
栈的图像展示

例如我们有一个存储整型元素的栈，我们依次压栈：{1, 2, 3}



在压栈的过程中，栈顶的位置一直在“向上”移动，而栈底是固定不变的。

如果我们要把栈中的元素弹出来：



出栈的顺序为3，2，1，顺序与入栈时相反，这就是所谓的“先入后出”。

在弹栈的过程中，栈顶位置一直在“向下”移动，而栈底一直保持不变。

栈的常用操作

弹栈：pop()

压栈：push()

求栈的大小：size()

判断栈是否为空：empty()

获取栈顶元素的值：top()

注意：栈无法像数组那样通过索引访问，只能访问栈顶元素

应用实例

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

```
using namespace std;
```

```
//输入字符串
```

```
string p;
```

```
//声明字符串栈
```

```
stack<char> s;

int main() {
    //清空栈
    while (!s.empty()) {
        s.pop();
    }
    cin >> p;
    //将字符压入栈
    for (int j = 0; j < p.size(); ++j) {
        s.push(p[j]);
    }
    //逆序输出
    while (!s.empty()) {
        cout << s.top();
        s.pop();
    }
    return 0;
}
```

逻辑航线培优教育，信息学奥赛培训专家。

扫码添加作者获取更多内容。

