

线性结构

线性结构又称线性表，即每个元素只有一个唯一前驱，只有一个唯一后继。其中必定有一个唯一的“第一个元素”和唯一的“最后一个元素”

分类

- 顺序表
- 链表
 - 单链表
 - 双链表
 - 循环链表

动态数组

在STL中存在一种动态数组——`vector` 可以满足线性与动态存储的特点

std::vector

`vector` 的存储是自动管理的，按需扩张收缩。`vector` 通常占用多于静态数组的空间，因为要分配更多内存以管理以后的增长。`vector` 在额外内存耗尽时重分配。

`vector` 上的常见操作复杂度（效率）如下：

- 随机访问——常数 $O(1)$
- 在末尾插入或移除元素——均摊常数 $O(1)$
- 插入或移除元素——与到 `vector` 结尾的距离成线性 $O(n)$

使用示例

```
#include <iostream>
#include <vector>

int main()
{
    // 创建含有整数的 vector
    std::vector<int> v = {7, 5, 16, 8};

    // 添加二个整数到 vector
    v.push_back(25);
    v.push_back(13);

    // 迭代并打印 vector 的值
    for(int n : v) {
```

```
        std::cout << n << '\n';  
    }  
}
```

实现vector

详见作业