几乎所有 stl 类都有 reserve，这个函数可以用于真正释放空间和减少重复开内存带来的常数

几乎所有 stl 类都可以用 for each 语法，即 for (auto x: a)，但是需要注意，for 的过程中不可对这个类有大小上的修改，一个经典例子是在图上 dfs 时，递归 修改了当前正在使用的数组导致 UB。

几乎所有 stl 类都有迭代器，可以类似于指针，end 类似于空指针。注意 begin end 是左闭右开，而 front back 均是闭。

```
bool operator < (Node const& _A) const
```

iterator

vector

- 不要使用 vector
- size
- clear
- empty
- push_back(尾部添加一个元素), pop_back（尾部删除一个元素）
- resize（把容器缩小和变大）
- 自带字典序比较

string

- 与 vector 类似
- substr（pos, size）(复杂度 size)
- find（复杂度 nm）
- +=

set/multiset/map(multimap 通常不用)

在set中每个元素的值都唯一，而且系统能根据元素的值自动进行排序 multiset 是关联容器的一种，是排序好的集合（元素已经进行了排序），并且允许有相同的元素。 要修改 multiset 容器中某个元素的值，正确的做法是先删除该元素，再插入新元素 map 第一个可以称为关键字(key)，每个关键字只能在map中出现一次；第二个可能称为该关键字的值(value)

- clear
- insert (复杂度 log)
- count 查找set中某个某个键值出现的次数 注意 multiset 复杂度有问题，如有需求使用 map (set map 复杂度都是log， multiset 用find)
- find 返回给定值值得迭代器，如果没找到则返回end() （复杂度 log）
- lower_bound 返回第一个大于等于key_value的定位器 要找值应该写*s.lower_bound(x) 但要在前面判断等不等于s.end()
- upper_bound 返回最后一个大于等于key_value的定位器
- erase(x) 指删除所有x 在multiset中如果要删除一个 应该写s.erase(s.find(x))

priority_queue

- 默认大根(降序) 小根堆：priority_queue <int,vector,greater > q
- 没有 clear
- push 插到队尾并排序
- pop 弹出队头
- empty
- top

stack(可用Vector) queue（尽量不用） 尽可能不要用 deque

使用 deque 数组时需要注意空间问题 当需要向序列两端频繁的添加或删除元素时，应首选 deque 容器。

- front 第一个 back 最后一个
- push_back push_front
- pop_back pop_front

bitset

bitset <40> bst;

string s = "100101";或者 char ch[] = "100101" bitset<10>a3(s);//长度为10，前面用 0 补充

bitset<2>bitset1(12) 取前半部分 bitset2 取后半部分

- 各种位运算
- bitset 支持单点修改 bst[23] = 1
- _Find_first(找到从低位到高位第一个1的位置), _Find_next(找到当前位置的下一个1的位置)(复杂度为距离/64)
- count 用来求bitset中1的位数

unordered_set, unordered_map

- 自定义哈希 （unordered_set < string > 或者自定义结构体都会报错）
- map过得去的题就不要unordered

常用内置函数

- 数学类
- sort, stable_sort(保证相等元素的相对位置), merge(2 个有序序列合并为 1 个有序序列), unique(cnt = unique(a + 1, a + 1 + n) - a - 1)
- nth_element(a,a+k,a+n)，函数只是把下标为k的元素放在了正确位置，对其它元素并没有排序，当然k左边元素都小于等于它，右边元素都大于等于它，所以可以利用这个函数快速定位某个元素(复杂度O(n))
- next_permutation(a + 1, a + 1 + n)
- memset, memcpy（多测使用须谨慎） memcpy(b, a,sizeof(b))
- strlen, strcmp
- swap（所有 stl 类 swap 均是 O(1)）