vector

c.front() 返回第一个数据 O(1)
c.back() 返回数组中的最后一个数据 O(1)
c.pop_back() 删除最后一个数据 O(1)
c.resize(n, v) 改变数组大小为 n,n 个空间数值赋为 v，如果没有默认赋值为 0
c.insert(it, x) 向任意迭代器 it 插入一个元素 x, O(N)
c.erase(first,last) 删除[first,last)的所有元素， O(N)
c.empty() 判断是否为空，为空返回真，反之返回假 O(1)

stack

s.push(ele) 元素 ele 入栈，增加元素 O(1)
s.pop() 移除栈顶元素 O(1)
s.top() 取得栈顶元素（但不删除）O(1)
s.empty()检测栈内是否为空，空为真 O(1)
s.size() 返回栈内元素的个数 O(1)

queue

q.front() 返回队首元素 O(1)
q.back() 返回队尾元素 O(1)
q.push(element) 尾部添加一个元素 element 进队 O(1)
q.pop() 删除第一个元素 出队 O(1)
q.size() 返回队列中元素个数，返回值类型 unsigned intO(1)
q.empty() 判断是否为空，队列为空，返回 true O(1)

deque

push_back(x)/push_front(x) 把 x 插入队尾后 / 队首 O(1)
back()/front() 返回队尾 / 队首元素 O(1)
pop_back() / pop_front() 删除队尾 / 队首元素 O(1)
erase(iterator it) 删除双端队列中的某一个元素
erase(iterator first,iterator last) 删除双端队列中[first,last)中的元素
empty() 判断 deque 是否空 O(1)
size() 返回 deque 的元素数量 O(1)
clear() 清空 deque

map

mp.find(key) 返回键为 key 的映射的迭代器 $O(logN)$ 注意：数据不存在时，返回 mp.end()
mp.erase(it) 删除迭代器对应的键和值 O(1)
mp.erase(key)根据映射的键删除键和值 O(logN)

mp.erase(first,last) 删除左闭右开区间迭代器对应的键和值 O(last−first)

mp.size() 返回映射的对数$ O(1)$
mp.clear() 清空 map 中的所有元素 O(N)
mp.insert() 插入元素，插入时要构造键值对

mp.empty()　如果 map 为空，返回 true，否则返回 false
mp.begin()　返回指向 map 第一个元素的迭代器（地址）
mp.end()　返回指向 map 尾部的迭代器（最后一个元素的下一个地址）
mp.rbegin()　返回指向 map 最后一个元素的迭代器（地址）
mp.rend()　返回指向 map 第一个元素前面(上一个）的逆向迭代器（地址）
mp.count(key)　查看元素是否存在，map 中键是唯一的，存在返回 1，不存在返回 0
mp.lower_bound()　返回一个迭代器，指向键值>= key 的第一个元素
mp.upper_bound()　返回一个迭代器，指向键值> key 的第一个元素

set

s.begin() 返回 set 容器的第一个元素的地址（迭代器）O(1)
s.end()　返回 set 容器的最后一个元素的下一个地址（迭代器）O(1)
s.rbegin()返回逆序迭代器，指向容器元素最后一个位置 O(1)
s.rend()　返回逆序迭代器，指向容器第一个元素前面的位置 O(1)
s.clear()　删除 set 容器中的所有的元素,返回 unsigned int 类型 O(N)
s.empty()判断 set 容器是否为空 O(1)
s.insert() 插入一个元素
s.size()　返回当前 set 容器中的元素个数 O(1)
erase(iterator)　删除定位器 iterator 指向的值
erase(first,second)　删除定位器 first 和 second 之间的值
erase(key_value)　删除键值 key_value 的值
s.find(element)　查找 set 中的元素，有则返回该元素对应的迭代器，无则返回结束迭代器
s.count(element)　查询 element 是否出现
s.lower_bound(k)　返回大于等于 k 的第一个元素的迭代器 O(logN)
s.upper_bound(k)　返回大于 k 的第一个元素的迭代器 O(logN)

string

s.insert(pos,element)　在 pos 位置插入 element
s.append(str)　在 s 字符串结尾添加 str 字符串
erase(iterator p)　删除字符串中 p 所指的字符
erase(iterator first, iterator last)　删除字符串中迭代器区间[first,last)上所有字符
erase(pos, len)　删除字符串中从索引位置 pos 开始的 len 个字符
s.replace(pos,n,str)把当前字符串从索引 pos 开始的 n 个字符替换为 str
s.replace(pos,n,n1,c)　把当前字符串从索引 pos 开始的 n 个字符替换为 n1 个字符 c
s.replace(it1,it2,str)把当前字符串[it1,it2)区间替换为 str it1 ,it2 为迭代器哦
s.substr(pos,n)　截取从 pos 索引开始的 n 个字符
s.find (str, pos)　在 pos 索引位置开始查找子串 str，返回找到的位置索引，-1 表示找不到
s.find (c, pos)　在 pos 索引位置开始查找字符 c，返回找到的位置索引，-1 表示找不到
s.rfind (str, pos)　反向查找子串 s，返回找到的位置索引，-1 表示查找不到子串
s.rfind (c,pos)　反向查找字符 c，返回找到的位置索引，-1 表示查找不到字符
s.find_first_of (str, pos) 在当前字符串的 pos 索引位置（默认为 0）开始，查找子串 s 的字符
s.find_first_not_of (str,pos)　在 pos 索引位置开始查找第一个不位于子串 s 的字符
s.find_last_of(str, pos)　在 pos 索引位置开始查找最后一个位于子串 s 的字符
s.find_last_not_of (str, pos) 在在 pos 索引位置开始查找最后一个不位于子串 s 的字符