

函数原型:

- `set_intersection(iterator beg1, iterator end1, iterator beg2, iterator end2, iterator dest);`

// 求两个集合的交集

// 注意:两个集合必须是有序序列

// beg1 容器1开始迭代器

// end1 容器1结束迭代器

// beg2 容器2开始迭代器

// end2 容器2结束迭代器

// dest 目标容器开始迭代器

`vector<int>vTarget;`

// 目标容器需要提前开辟空间

// 最特殊情况 大容器包含小容器 开辟空间 取小容器的size即可

`vTarget.resize(min(v1.size(), v2.size()));`

main 算法 取较小值

// 获取交集

`vector<int>::iterator itEnd = set_intersection(v1.begin(), v1.end(), v2.end(), vTarget.begin());`

总结:

- 求交集的两个集合必须的有序序列
- 目标容器开辟空间需要从两个容器中取小值
- `set_intersection`返回值既是交集中最后一个元素的位置