**Standard Template Library**

1. **容器**

1.1 pair x.first x.second

1.2 vector deque list set multiset map multimap

*multi-sm[set map] veld [vector ? list deque] 在无树平原上搞多人SM☹*

1. **常用容器multi-sm veld**

2.1 通用方法

x::value\_type

x::iterator x::const\_iterator

x::reverse\_iterator x::const\_reverse\_iterator

=

x::begin x::end

x::rbegin x::rend

x::size x::max\_size x::empty x::clear x::swap

Comparison Operators <>=

*Be[begin end] common cases[x::clear x::max\_size x::size x::empty x::swap] 成为常规案例*

2.2 序列化容器Sequence Containers

veld[vector deque list]

insert erase push\_back pop\_back front back reverse

*2pb fibers[front insert back erase reverse swap] 对两个pb纤维排序*

2.3 vector []

2.4 deque

[] push\_front pop\_front

*deque 2pf[push\_front pop\_front]*

2.5 list

pop\_front push\_front splice remove remove\_if uniq merge sort

*remove 2pf[push\_front pop\_front] sums[splice uniq merge sort] 移除两个pf的和*

2.6 排序的关联性容器multi-sm

set multiset map multimap

insert erase count find lower\_bound upper\_bound equal\_range

*b-lur[-bound lower upper equal\_range] fiche[find insert count ? erase] 模糊的胶卷*

1. **容器适配器**

stack queue priority-queue

*stack priority-queue对优先队列堆栈*

3.1 Stack Adaptor <vector deque list>

empty size push pop top

3.2 queue adaptor <deque list>

empty size push pop front back

3.3 priority-queue <vector deque>

empty size push pop top

*bf`s[front back size] top pep[empty push pop] 男朋友的最好精力*

1. **算法**

4.1 查询算法Query Algorithms

for\_each mismatch equal search adjacent\_find random\_shuffle partition stable\_partition

find count

find\_if count\_if

*for-each stable\_partition, find & count smear[search mismatch equal adjacent\_find random\_shuffle] 对每一个稳定的分区, 查找和计数污点*

4.2 变异算法Mutating Algorithms

copy copy\_backward swap\_ranges transform

replace replace\_copy remove remove\_copy

replace\_if replace\_copy\_if remove\_if remove\_copy\_if

fill generate

fill\_n generate\_n

uniq reverse rotate

uniq\_copy reverse\_copy rotate\_copy

*copy-backward 4uric[uniq replace remove reverse rotate -if -copy] gifts[generate \* fill transform swap\_ranges] 四个沾尿的礼物*

4.3 排序及其应用Sort and Application

sort stable\_sort prtial\_sort partial\_sort\_copy nth\_element

*sort pens[prtial\_sort nth\_element nth\_element stable\_sort] 对笔排序*

4.3.1 Binary Search

binary\_search lower\_bound upper\_bound equal\_range

*blur[binary\_search lower\_bound upper\_bound equal\_range] 模糊搜索*

4.3.2 Merge

merge inplace\_merge

*inplace-merge 在位合并*

4.3.3 Functions on set

includes set\_union set\_intersection set\_difference set\_symmetric\_difference

*includes set-dui[堆set\_union set\_intersection set\_difference] 包含set堆*

4.3.4 堆HEAP

push\_heap pop\_heap make\_heap sort\_heap

*sort-pump-heap[sort\_heap push\_heap make\_heap pop\_heap] 对泵堆排序*

4.3.5 Min and Max

min max min\_element max\_element

*max-element 最大元素*

4.3.6 置换Permutations

next\_permutation prev\_permutation

*np-permutation NP置换*

4.4 Computational

accumulate inner\_product partial\_sum ajacent\_difference

*paid[partial\_sum accumulate inner\_product ajacent\_difference] 已经付过钱*

**String**

1. **容量**

size length max\_size resize capacity reserve clear empty shrink\_to\_fit;

operator[] at back front begin end;

*be[begin end] bf[front back] case[clear max\_size size empty swap] 变成男朋友的案例*

1. **修改**

operator+= append push\_back assign insert erase replace swap pop\_back;

length assign replace;

*alarm[assign length append replace max\_size] ! 2pb[push\_back pop\_back] fibers [front insert back erase reverse swap] 警告! 2根pb光纤.*

1. **字符串操作**

c\_str data get\_allocator copy find rfind find\_first\_of find\_last\_of find\_first\_not\_of find\_last\_not\_of substr compare;

*find-first-not-of c-str data, copy n[npos] scarfs[substr compare at rfind find swap] 找到第一个不是c\_str的数据, 拷贝N条围巾*

1. **非成员函数**

*gs[getline swap] GS段*

**Fstream**

1. **公开成员函数**

open is\_open close rdbuf operator= swap

*swap rico[rdbuf is\_open close open]*

1. **从istream继承来的公开成员函数**

operator>> gcount get getline ignore peek read readsome putback unget tellg seekg sync;

get read tellg seekg sync;

operator<< put write tellp seekp flush

*gcount getline;*

*string[seekg seekp tellg tellp read ignore narrow get] pews[put eof write sync] 绳子做的长椅*

1. 从ios继承来的公开成员函数

good eof fail bad operator! operator bool rdstate setstate clear copyfmt fill exceptions imbue tie rdbuf narrow widen