

# 10 minutes de visite de la STL

et al.

Chris Xiong

2018-05-04

# 10 minutes de visite de la STL (et al.)

## Outline

- ▶ STL
- ▶ Faster I/O
- ▶ C++11 / 14
- ▶ Common pitfalls
- ▶ 如何让队友认错之如何表扬队友

# 10 minutes de visite de la STL (et al.)

## STL

- ▶ Containers
- ▶ Algorithms
- ▶ Utilities
- ▶ Strings
- ▶ Iterators
- ▶ IO
- ▶ Extensions
- ▶ ~~RegExp, Threading, etc.~~

# 10 minutes de visite de la STL (et al.)

## STL - Containers

- ▶ Sequential
- ▶ Associative
- ▶ Unordered associative
- ▶ Adaptors
- ▶ Function table
- ▶ Iterator invalidation

# 10 minutes de visite de la STL (et al.)

STL - Containers - Sequential

- ▶ `vector`
- ▶ `deque`
- ▶ `list`
- ▶ `array`
- ▶ `forward_list`

# 10 minutes de visite de la STL (et al.)

## STL - Containers - Associative

- ▶ tree-based,  $O(\log n)$  for most operations.
- ▶ (like TreeMap in java)
- ▶ set: maps from something to bool (existence).
- ▶ map: "array" whose subscript can be anything ordered.
- ▶ multiset
- ▶ multimap

# 10 minutes de visite de la STL (et al.)

## STL - Containers - Unordered associative

- ▶ **C++ 11 only**, hash-based,  $O(1)$  for most operations.
- ▶ (like HashMap in java)
- ▶ `unordered_set`
- ▶ `unordered_map`: "array" whose subscript can be anything hashable.
- ▶ `unordered_multiset`
- ▶ `unordered_multimap`

# 10 minutes de visite de la STL (et al.)

STL - Containers - Adaptors

- ▶ `stack`
- ▶ `queue`
- ▶ `priority_queue`



# 10 minutes de visite de la STL (et al.)

## STL - Containers - Function table

	vector	deque	list	set	map	unordered_set	unordered_map	stack	queue	priority_queue	
ordered?				✓	✓					✓	
iterators (begin/end)	✓	✓	✓	✓	✓	✓	✓				
at/[]	✓	✓			✓		✓				
size	✓										
clear	✓										
insert	✓										
erase	✓										
push_front			✓								
pop_front			✓							pop	pop
front		✓								✓	top
push_back		✓							push	push	push
pop_back		✓							pop		
back		✓							top	back	

Complete version [here](#).

# 10 minutes de visite de la STL (et al.)

## STL - Containers - Iterator invalidation

- ▶ Very common pitfall that results in RE.
- ▶ Associative containers: always valid after insertion/deletion.
- ▶ Unordered associative containers: valid until rehash.
- ▶ Sequential: invalid after relocation.
- ▶ Also STL throws exceptions, beware.

# 10 minutes de visite de la STL (et al.)

## STL - Algorithms

- ▶ `sort`
- ▶ `stable_sort`
- ▶ `reverse`
- ▶ `unique`
- ▶ `lower_bound`, `upper_bound`
  - ▶ Also a member function of ordered associative containers.
- ▶ `make_heap`, `push_heap`, `pop_heap`, `sort_heap`
- ▶ `min`, `max`
- ▶ `next_permutation`, `prev_permutation`

# 10 minutes de visite de la STL (et al.)

## STL - Utilities

- ▶ pair
- ▶ tuple

# 10 minutes de visite de la STL (et al.)

## STL - Strings

- ▶ like a vector
- ▶ `substr`
- ▶ `find`, `replace`
  - ▶ `find` very fast in the implementation we use!

# 10 minutes de visite de la STL (et al.)

## STL - Extensions

- ▶ `pb_ds`
  - ▶ `__gnu_pbds::tree`
  - ▶ `__gnu_pbds::trie`
- ▶ `__gnu_cxx::rope`
  - ▶ Long strings
  - ▶ Very fast substr and concat ( $O(\log n)$  compared to  $O(n)$  in normal string)
  - ▶ Slower random access ( $O(\log n)$  compared to  $O(1)$  in normal string)

# 10 minutes de visite de la STL (et al.)

## STL - Quick examples

```
std::vector<int> v;//vector of int
v.push_back(1);//append 1 to the vector
v[0]=2;//access the first element of the vector
std::sort(v.begin(),v.end());//sort the vector
std::sort(v.begin(),v.end(),std::greater<int>());//sort the vector in an alternative order
std::upper_bound(v.begin(),v.end(),0);//locate the first value greater than 0 in the (now sorted) vector
for(size_t i=0;i<v.size();++i)...//iterate through the vector
for(auto& i:v)...//same as above but using for each loop, value of i equals to the current element

std::set<int> s;//set of int
s.insert(1);//add 1 to the set
s.find(1);//returns the iterator pointing to 1 in the set
s.find(2);//returns s.end()
for(auto& i:s)...//iterate through the set in its order, use i to access the value
s.lower_bound(0);//finds first element greater or equal to 0 in the set, returning the iterator pointing to it
s.erase(1);//erase 1 from the set

std::map<std::string,int> m;//map from string to int
m["abc"]=1;//access an element of the map, creating it if not exist. In this case assigning value 1 to key "abc".
m.find("abc");//returns an iterator
m.find("efg");//returns m.end()
for(auto& i:m)...//iterate through the map in its order, use i.first to access the key, i.second to access the value
m.erase("abc");//erase value with key "abc"
```

# Faster I/O

Why?

- ▶ No need to parse the format string
- ▶ Makes better use of the I/O buffer



# Faster I/O

Why?

- ▶ No need to parse the format string
- ▶ Makes better use of the I/O buffer

How?

- ▶ `ios_base::sync_with_stdio(false)`
- ▶ `getchar()`
- ▶ `fread()`

# Faster I/O

## Faster input - getchar()

```
int readint()
{
    int ret=0;
    char ch=getchar();
    while(ch<'0' || ch>'9')
        ch=getchar();
    do
    {
        ret=ret*10+ch-'0';
        ch=getchar();
    }while(ch>='0' && ch<='9');
    return ret;
}
```

# Faster I/O

Even faster input - fread()

```
char s[SOME_LARGE_VALUE];
size_t ptr,len;
len=fread(s,1,SOME_LARGE_VALUE,stdin);
int readint()
{
    //replace getchar() with s[ptr++]
    ...
}
```

# Faster I/O

Even faster input - fread()

```
char s[SOME_LARGE_VALUE];
size_t ptr,len;
len=fread(s,1,SOME_LARGE_VALUE,stdin);
int readint()
{
    //replace getchar() with s[ptr++]
    ...
}
```

Reading real numbers?

# Faster I/O

## Faster output

- ▶ Print digit by digit
- ▶ Save to buffer then `fwrite()` to `stdout`.

# C++11 / 14 - Useful features that reduces coding efforts

C++11

- ▶ Uniform initialization

```
struct S{int a;double b;char c;};  
S s{1,2.0,'3'};//s.a==1; s.b==2.0; s.c=='3';
```

- ▶ Type inference

```
std::set<int> s;  
auto a=1;  
auto it=s.begin();
```

- ▶ Range-based for loop

```
std::vector<int> v;  
for(auto &i:v)...
```

## C++11 / 14 - Useful features that reduces coding efforts

### C++11 cont'd

- ▶ Lambda functions

```
std::sort(p+1,p+nv,[o=p[0]](const v3d& a,const v3d& b)->bool
        {
            double cross=(a.x-o.x)*(b.y-o.y)-(b.x-o.x)*(a.y-o.y);
            if(sgn(cross))return sgn(cross)>0;
            double la=hypot(a.x-o.x,a.y-o.y);
            double lb=hypot(b.x-o.x,b.y-o.y);
            return la<lb;
        }
    );
```

- ▶ Right angle bracket

```
std::vector<std::pair<int,int>> v;
```

- ▶ long long int

Yeah, it was not standard until C++11 (C99).

# C++11 / 14 - Useful features that reduces coding efforts

## C++11 STL

- ▶ Hash tables  
`std::unordered_map`, `std::unordered_set`
- ▶ `std::tuple`
- ▶ Regular expressions
- ▶ Threading
- ▶ Random number generators



# C++11 / 14 - Useful features that reduces coding efforts

## C++14

C++14 is a small extension to C++11.

- ▶ Improved auto

```
auto factorial(int x){  
    if(x==1)return 1;  
    return x*factorial(x-1);  
}
```

- ▶ Improved lambda functions

```
auto lambda=[](auto x,auto y){return x+y};
```

- ▶ Lambda capture expressions

## Common pitfalls

- ▶ Huge global variable causes linkage error.

```
int a[1LL<<12][1LL<<48], b[1LL<<12][1LL<<48], c[1LL<<12][1LL<<48];  
int main(){c[0][0]=1;}
```

Compilation result:

```
g++ -Wall -std=c++14 -g -o "test" "test.cpp" -lm (in directory:  
/home/chrisoft/code)
```

```
/tmp/cc0kFQ7A.o: In function 'main':
```

```
/home/chrisoft/code/test.cpp:2:(.text+0x6):
```

```
relocation truncated to fit: R_X86_64_PC32 against symbol  
'c' defined in .bss section in /tmp/cc0kFQ7A.o
```

```
collect2: error: ld returned 1 exit status
```

```
Compilation failed.
```

- ▶ relocation of .bss section exceeding platform limitations.

## Common pitfalls

- ▶ Stack size is very small compared to heap

```
int main(){int a[100000000];...}
```

- ▶ Results in stack overflow (even CE).

- ▶ Math library

- ▶ `memset()` TLE

- ▶ `strlen()` is  $O(n)$ !!! (however `std::string::length()` is  $O(1)$ )

# 如何让队友认错之如何表扬队友

队友需要表扬才能更有生产力



俞旭铮

( 你看 你也可以趁机表扬我一发对不对 > >

# 如何让队友认错之如何表扬队友

## 光辉事迹

- ▶ 比赛中要其他队伍帮忙写对拍
- ▶ 比赛中睡3小时觉
- ▶ 洒一地可乐
  - ▶ 下次灌崂山白花蛇草水
- ▶ 全权负责实验室事务
- ▶ 发说说

# 宇宙智障的说说

就终于考完试了\_(:з ∠)\_  
然而暑假这种? 不存在的(滑稽  
还是献给辣鸡的(划掉)acm好了  
学期结束,事儿反而就又突然多了起来  
假装已有计划的专题训练  
各种姿势被虐的多校联合  
以及暑期集训的萌新教学  
与开始正式接手的实验室各种大小事务

的锅  
相信,一定会是个忙碌的七八月份吧  
希望,也同会是个充满收获的七八月  
就也会更期待着  
在八月初的短假里  
与小姐姐愉快的玩耍呢  
总之,一定会是个不平凡的假期  
再以及,小姐姐寄的明信片也终于到了:)

# 宇宙智障的说说



俞旭铮 明天挂着这个上场,大概会有气势加成(滑稽脸)



拍摄于 2017年5月20日 西安市西北工业大学(长安校区)

2017年5月20日 来自小米5

赞(68) 评论(7) 转发 更多 ▾



# 为何要表扬队友

- ▶ 让恬不知耻的队友知道错（似乎对宇宙智障无效）
- ▶ 叫醒队友
- ▶ 鼓励队友WA更多题



# Thanks for listening

Slides will be available later.

Useful links:

- ▶ <http://www.cplusplus.com/reference/>
- ▶ <http://en.cppreference.com/w/>