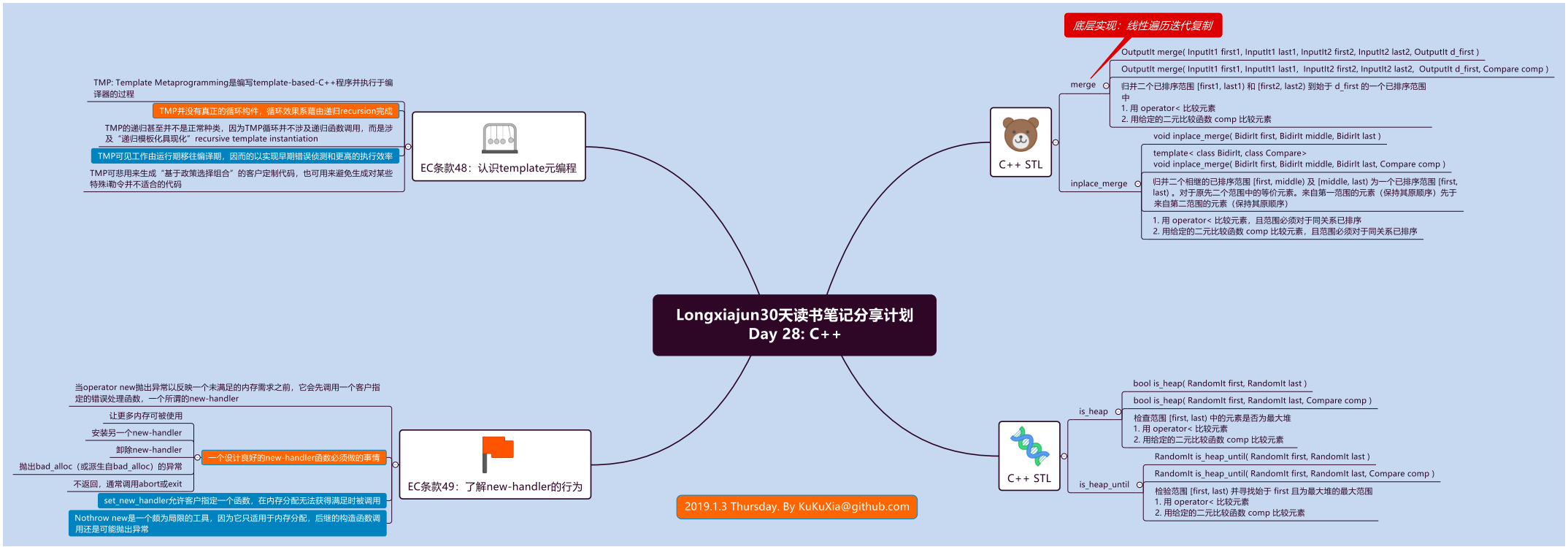
Longxiajun30天读书笔记分享计划
Day 28: C++
2019.1.3 Thursday. By KuKuXia@github.com
EC条款48：认识template元编程
TMP: Template Metaprogramming是编写template-based-C++程序并执行于编译器的过程
TMP并没有真正的循环构件，循环效果系藉由递归recursion完成
TMP的递归甚至并不是正常种类，因为TMP循环并不涉及递归函数调用，而是涉及“递归模板化具现化”recursive template instantiation
TMP可见工作由运行期移往编译期，因而的以实现早期错误侦测和更高的执行效率
TMP可悲用来生成“基于政策选择组合”的客户定制代码，也可用来避免生成对某些特殊i勒令并不适合的代码
EC条款49：了解new-handler的行为
当operator new抛出异常以反映一个未满足的内存需求之前，它会先调用一个客户指定的错误处理函数，一个所谓的new-handler
一个设计良好的new-handler函数必须做的事情
让更多内存可被使用
安装另一个new-handler
卸除new-handler
抛出bad_alloc（或派生自bad_alloc）的异常
不返回，通常调用abort或exit
set_new_handler允许客户指定一个函数，在内存分配无法获得满足时被调用
Nothrow new是一个颇为局限的工具，因为它只适用于内存分配，后继的构造函数调用还是可能抛出异常
C++ STL
底层实现：线性遍历迭代复制
merge
OutputIt merge(InputIt1 first1, InputIt1 last1, InputIt2 first2, InputIt2 last2, OutputIt d_first)
OutputIt merge(InputIt1 first1, InputIt1 last1, InputIt2 first2, InputIt2 last2, OutputIt d_first, Compare comp)
归并二个已排序范围 [first1, last1) 和 [first2, last2) 到始于 d_first 的一个已排序范围中
1. 用 operator< 比较元素
2. 用给定的二元比较函数 comp 比较元素
inplace_merge
void inplace_merge(BidirIt first, BidirIt middle, BidirIt last)
template< class BidirIt, class Compare>
void inplace_merge(BidirIt first, BidirIt middle, BidirIt last, Compare comp)
归并二个相继的已排序范围 [first, middle) 及 [middle, last) 为一个已排序范围 [first, last)。对于原先二个范围中的等价元素。来自第一范围的元素（保持其原顺序）先于来自第二范围的元素（保持其原顺序）
1. 用 operator< 比较元素，且范围必须对于同关系已排序
2. 用给定的二元比较函数 comp 比较元素，且范围必须对于同关系已排序
C++ STL
is_heap
bool is_heap(RandomIt first, RandomIt last)
bool is_heap(RandomIt first, RandomIt last, Compare comp)
检查范围 [first, last) 中的元素是否为最大堆
1. 用 operator< 比较元素
2. 用给定的二元比较函数 comp 比较元素
is_heap_until
RandomIt is_heap_until(RandomIt first, RandomIt last)
RandomIt is_heap_until(RandomIt first, RandomIt last, Compare comp)
检验范围 [first, last) 并寻找始于 first 且为最大堆的最大范围
1. 用 operator< 比较元素
2. 用给定的二元比较函数 comp 比较元素