

# LUT-XCPC Day 1 语法 & 基础数据结构

沈宇昊

兰州理工大学

July 19, 2022

# Contents

## 1. 前置

## 2. cpp98 → pp11

# 知识准备

基础的递归、模拟

基本时间复杂度会看，有概念

会写暴力

视情况而定，哪里需要补充哪里知识，临时讲一下即可

# 资料 & 训练方式

C++ Reference

OI Wiki

知乎

每场 CF 的题解 & 评论区，大佬的过题代码

上述有不懂的进一步百度 | Google、交流，主要是坚持做题

各种比赛省赛级别获奖对比赛经验和知识准备要求不高，当然之后如果甘肃有 ICPC 的省赛，和兰大同台竞技另说、

像 ZJ 等省份，XCPC 有省赛，拿金的难度可能还要大于区域银

# 题单

洛谷官方题单

2020,2021CF 简单题精选 适合有基础之后提高

搜索题单

关于做题和看题解.. 不会做看题解是正常的，尽量是一点点看，理解思路，然后尝试自己写出，不行再看代码，理解了之后再写，注意抄题解和看题解的区分，没有太多思考内容的题一般  
10-15min

# Contents

## 1. 前置

## 2. cpp98 → pp11

## summary

特定函数的部分主要看的是这篇 [洛谷日报](#)

模板，别的各种内容来自平时整理

## algorithm 库

常见的函数有 swap, sort, unique, reverse, lower\_bound, upper\_bound 等...

一些别的函数

std::find

std::fill 一般用来弥补 memset 不能赋值的问题

std::max\_element | min\_element (bg, ed) 第三个参数可传入比较函数

std::count(bg, ed, val)

std::count\_if(bg, ed, func) 常用 func 有 isdigit, islower, isupper 等

std::for\_each(bg, ed, func)



## numeric 库

```
std::accumulate(bg, ed, val)
```

可以用于序列求和，注意传参时 val 的类型避免溢出，第四个参数可以作为加法

```
std::partial_sum(bg1, ed1, bg2)
```

用于求前缀和，可以传入第四个参数作加法

```
std::adjacent_difference(bg1, ed1, bg2)
```

用于求差分，可以传入第四个参数作减法

## cmath 库

`exp(x)`

返回  $e^x$ ,  $x$  的有效范围是  $[-708.4, 709.8]$

`log(x)`

返回  $\ln x$ , 在  $x \leq 0$  时候报错, 别的还有  $\log_{10}$  和  $\log_2$ , 其中  $\log_2$  为 C++11 开始才有

`floor` 取上整

`ceil` 取下整

# GNU

这些内容不在 C++ 标准中，如 clang 等其他编译器里可能没有，一般比赛提供的编译器都是 GUN C++，也就是你们 dev 里面自带的 mingw(Minimalist GNU for Windows)

# GNU

## \_\_builtin 函数

\_\_builtin\_popcount(x) 统计二进制下 x 中 1 的个数

\_\_builtin\_parity(x) 统计二进制下 x 中 1 个数的奇偶性

\_\_builtin\_ffs(x) 统计二进制下最后一个 1 是从左往右第几位

\_\_builtin\_ctz(x) 返回二进制下后导 0 的个数

\_\_builtin\_clz(x) 返回二进制下前导 0 的个数