

软件基础实践

课程简介

上海交通大学软件学院

臧斌宇

背景

莫名奇妙的改革 — 工科大平台

- 工科学生前三个学期学习基本相同的课程
 - 高等数学、线性代数、物理及实验、化学及实验
 - 理论力学、电路理论及实验、程序设计、**数据结构**
 - 工程导论、工程实践
 - 概率统计、**离散数学**（数理方法）
 - 专业自设：**软件基础实践**、计算机系统基础（汇编）

背景

程序设计能力训练

- 软件基础实践（1学分）
- 算法设计与实现习题课（1学分）

软件基础实践（第二学期）

内容和方式

- **Lab**

- 课上布置和编码，TA随时解决问题
- 每次课当面检查上次课的Lab

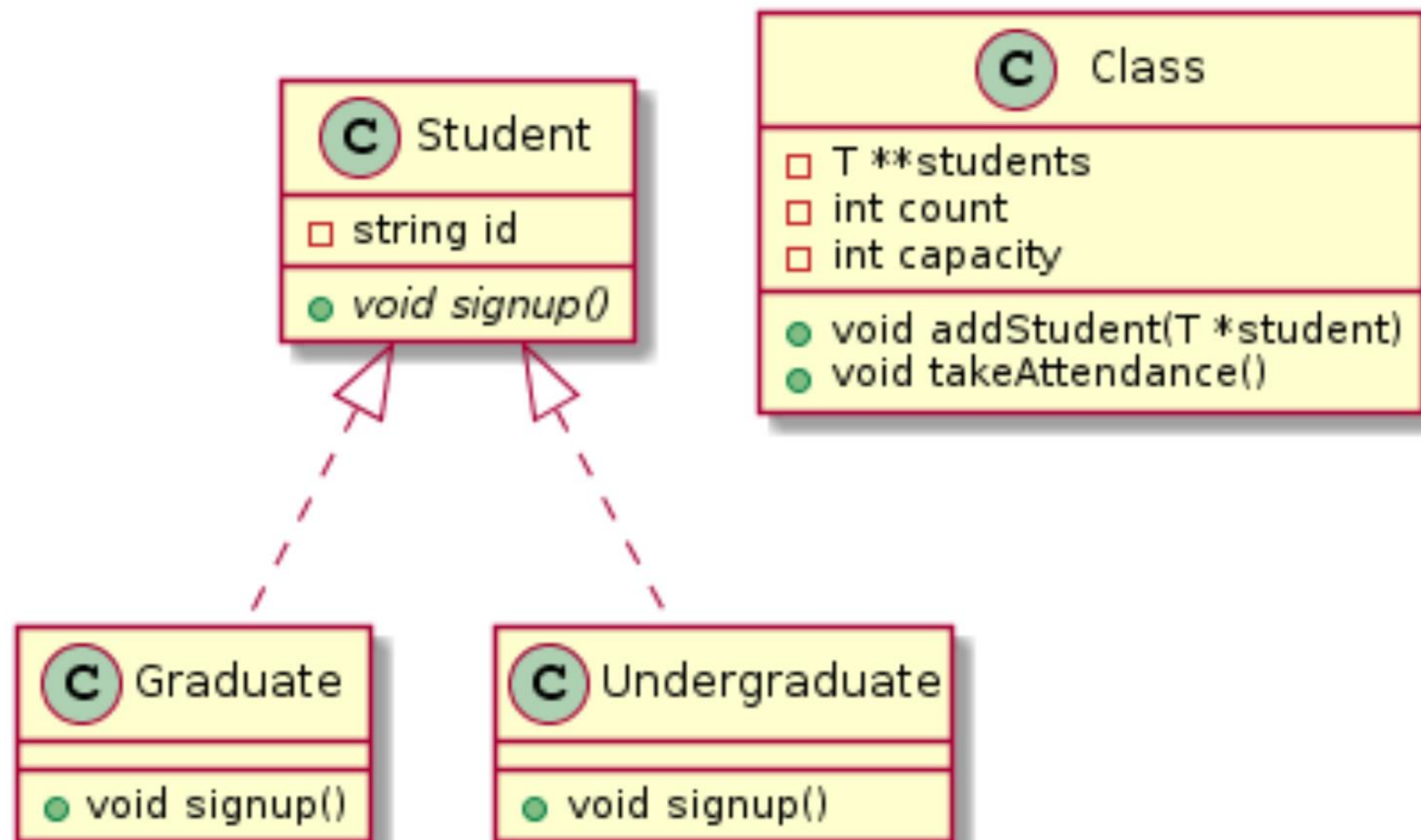
- **Project**

- 需要时间较长，线下实现
- 最后进行演示和答辩

Lab 1: 课程点名模拟

补充完成一个简单的C++程序

- 目的
 - 复习第一学期C++课程内容
 - 熟悉C++的集成开发环境（VS）
- 涵盖内容
 - C++字符串操作、输出、函数调用
 - 类、继承、虚函数、模板



Lab 2: MiniEd 行文本编辑器

- Ed是Ken Thompson开发的一个非常经典的行文本编辑器
 - 被用于文本文件的创建、显示、更改和其他一些操作
- 两种模式
- command mode
 - 启动时的模式
 - 输入的文本会被当做命令进行处理
- input mode
 - 通过一些特殊命令切换到输入模式
 - 输入的文本会被当做文件内容进行保存

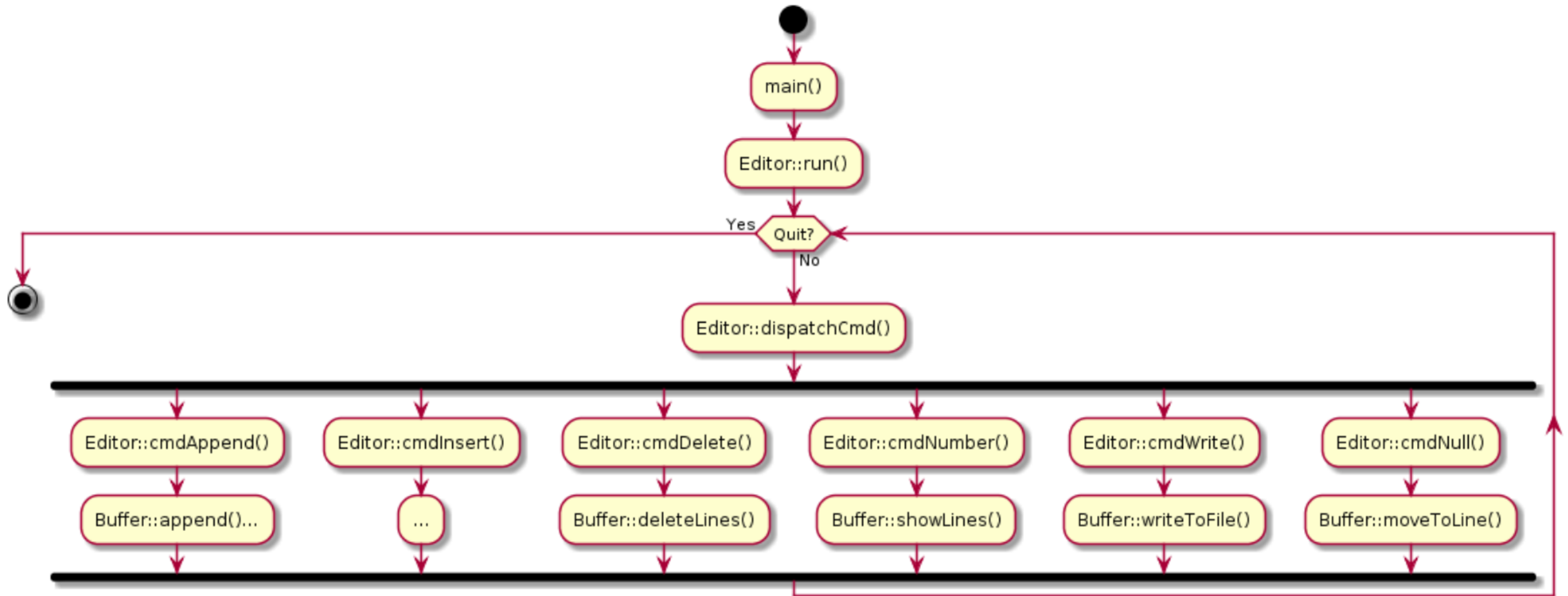


```
$ ed fstab
Newline appended
116
%1
/dev/hda2 / ext2 defaults 1 1\r$
/dev/hdb1 /home ext2 defaults 1 2\r$
/dev/hda1 swap swap pri=40 0 0$
3s/40/42/
w fstab
116
-
```

Lab 2: MiniEd 行文本编辑器

- MiniEd实现Ed命令的一个子集
 - 进一步熟悉C++的集成开发环境
 - 理解提供的C++代码，并根据需求增加相应功能
- 涵盖内容
 - C++字符串操作（根据给定格式进行解析）
 - C++标准输入输出流、文件输入输出流的使用
 - 类、继承
 - 异常处理（如非法输入）

Lab 2: MiniEd 行文本编辑器



Lab 3: 基于链表 (List) 的MiniEd

```
cmd> 100 let a = b + c
cmd> 30  if T < 0 then 70
cmd> 70  printf "hello world\n"
cmd> 30  printf "Override an existing line\n"
cmd> list
cmd> List

#For the above inputs, the Eidtor will print:
30  printf "Override an existing line\n"
70  printf "hello world\n"
100 let a = b + c
```

```
cmd> d 100
cmd> d 9999

cmd> save code.txt
cmd> save hello
```

Lab 3: 基于链表 (List) 的MiniEd

使用链表实现文本编辑器的数据存储部分

- 目的
 - 修改面向对象程序中的某个部件
 - 复习和掌握C++中的指针
- 涵盖内容
 - 链表的类设计、指针操作
 - 动态内存管理 (new、delete)

TUI: Text-based User Interface

```
.config - Linux/x86 4.17.3 Kernel Configuration

Linux/x86 4.17.3 Kernel Configuration
Arrow keys navigate the menu. <Enter> selects submenu ---> (or empty submenu ----). Highlighted
letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc>
to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module <> module
capable

[*] 64-bit kernel
  General setup --->
  [*] Enable loadable module support --->
  [*] Enable the block layer --->
  Processor type and features --->
  Power management and ACPI options --->
  Bus options (PCI etc.) --->
  Executable file formats / Emulations --->
[*] Networking support --->
  Device Drivers --->
  Firmware Drivers --->
  File systems --->
  Kernel hacking --->
  Security options --->
  -* Cryptographic API --->
[*] Virtualization --->
  Library routines --->

<Select> < Exit > < Help > < Save > < Load >
```

TUI 例1: Linux kernel menuconfig

```
Tabs
Tab0 | Tab1 |

Graphs
Gauge:
Sparkline:
List
> Item1
Item2
Item3
Item4
Item5
Item6
Item7
Item8
Item9
Item10
List
INFO: Event9
INFO: Event10
CRITICAL: Event11
INFO: Event12
INFO: Event13
INFO: Event14
INFO: Event15
INFO: Event16
ERROR: Event17
ERROR: Event18

Bar chart
Footer
This is a paragraph with several lines.
You can change the color.
Use {fg=[color];bg=[color];mod=[modifier]} [text] to highlight the text with a color. For example, under the rainbow.
Oh, and if you didn't notice you can automatically wrap your text =).
One more thing is that it should display unicode characters properly: ☺ ☻ ☼ ☽ ☾ ☿ ♀ ♂ ♁ ♃ ♄ ♅ ♆ ♇ ♈ ♉ ♊ ♋ ♌ ♍ ♎ ♏ ♐ ♑ ♒ ♓ ♔ ♕ ♖ ♗ ♘ ♙ ♚ ♛ ♜ ♝ ♞ ♟ ♠ ♡ ♢ ♣ ♤ ♥ ♦ ♧ ♨ ♩ ♪ ♫ ♬ ♭ ♮ ♯ ♰ ♱ ♲ ♳ ♴ ♵ ♶ ♷ ♸ ♹ ♺ ♻ ♼ ♽ ♾ ♿
```

TUI 例2: <https://github.com/fdehau/tui-rs>

Lab 4: Tower of Hanoi 汉诺塔

实现一个字符界面可视化汉诺塔游戏

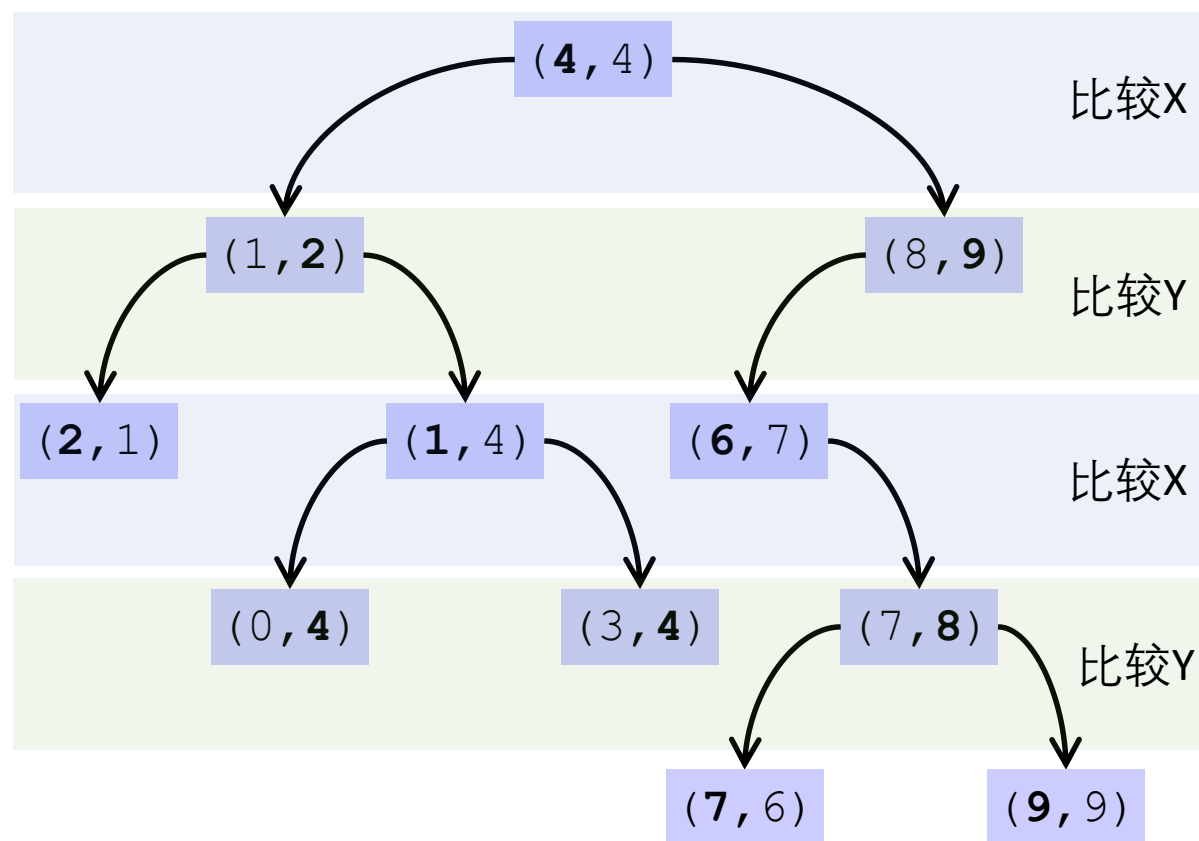
- 目的
 - 回顾和练习递归
 - 练习字符界面交互 (TUI) 的使用
- 涵盖内容
 - 递归求解
 - 求解过程的自动化展示

```
./hanoi
How many disks do you want? (1 ~ 5)
4
|
| *****
| *****
| *****
| *****
|-----|-----|-----|
|
|
| *****
| *****
| *****
| *****
|-----|-----|-----|
Move a disk.Format: x y
1 2
|
| *****
| *****
| *****
| *****
|-----|-----|-----|
```

学生作品

Lab 5: 2D-Tree

- 按照要求实现一个2D树结构
- 支持快速查询某点的最近点



From Stanford CS106B

Lab 5: 2D-Tree^[1]

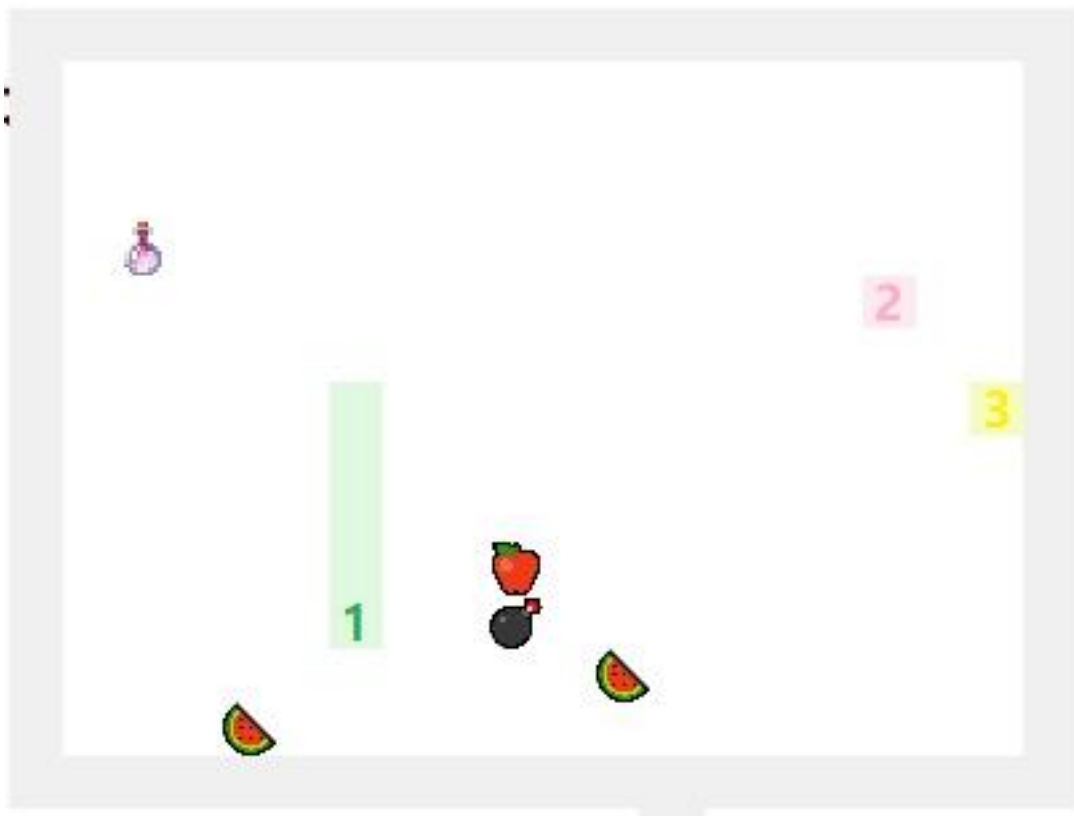
- 目的
 - 练习复杂数据结构的实现（2D-树）
 - 强化指针的使用
 - 练习给定算法的实现
- 涵盖内容
 - 容器类的设计
 - 树的基本操作（增删改查）
 - 指针使用
 - 动态内存管理（new、delete）
 - 基于2D树快速查找最近点

Project 1: QSnake 贪吃蛇

使用Qt图形库设计和实现一个图形化贪吃蛇游戏

- 目的
 - 学习和掌握图形界面交互(GUI) (使用Qt)
 - 练习面向对象设计
 - 简单的AI算法实现 (蛇自动寻食物)
- 涵盖内容
 - 类的设计和继承、多态的使用
 - 动态内存管理
 - 图形化显示和交互 (显示图形/图片、事件响应和处理)
 - 深度优先/广度优先搜索

学生作品



- #1 Adam 8
- #2 Doug 6
- #3 Ben 5
- #4 James 5
- #5 Eve 4
- #6 Hari 3
- #7 Harry 3
- #8 mysnake 3
- #9 Hulk 2
- #10 Elvin 1

Your Score : 3
Position : 8 out of 10

Elvin normal

mysnake

Hulk chipku

Ben courage

Adam paytu

Harry paytu

Hari normal

James chipku

Eve chipku

Potter paytu

软件基础实践（第三学期）

方式

- 上课介绍Project（两次）
- Lab
 - 只有一次课
 - 讲解带回溯的递归（八皇后）
 - 练习带回溯的递归（全排列）

Project 2: Boggle找单词游戏

根据给出的Qt显示框架代码，完成一个找单词的游戏

- 目的
 - 练习递归和回溯算法的设计
 - 熟悉图形化用户交互和事件处理
- 涉及内容
 - C++文件操作、字符串操作
 - 指针的使用
 - 数据结构的使用（链表、字典树（Trie）等）
 - 递归和回溯
 - 事件处理

Modified from Stanford CS106B

学生作品

QBoggle!

Me 5

stand rats sate
feet

W	Z	E	F	T
L	E	S	T	N
E	U	T	A	N
O	D	R	D	O
A	E	R	I	H

Computer 519

weet	weets	wees
weest	west	wets
weel	weed	weeder
zeta	zetas	zest
zees	efts	etna
etnas	etas	estate
feta	fetas	fets
fees	feel	fezes
lezes	leet	leets
lees	lest	lets
leud	lues	lust

Welcome to the game Boggle!
stand
rats
nado
noda
sate
zets
feet

Now it's the computer's turn!!
You lose,human
|

Project 3: Minimal Basic

To: BASIC Development Group
From: Bill Gates
Subject: C++ reimplementation
Date: April 1, 1981

This guy from Bell Labs, Bjarne Stroustrup, just sent me a parser program written in a new language he's calling C++. His parser is much simpler than ours but still seems quite efficient. The code is much easier to read as well.

I think it's time to move away from assembly language for our version of BASIC, and C++ may be just the right tool.

Please get going on this project as soon as possible.

Bill Gates

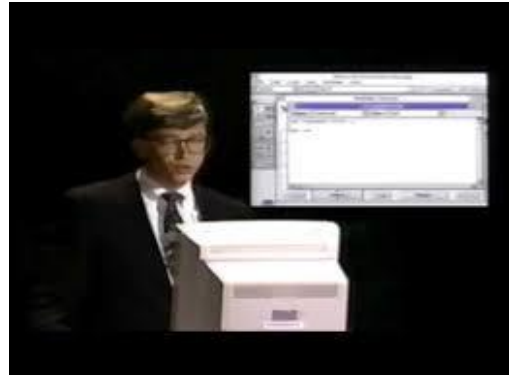
```
Basic
Minimal BASIC -- Type HELP for help.

100 REM Program to print the Fibonacci sequence
110 LET max = 10000
120 LET n1 = 0
130 LET n2 = 1
140 IF n1 > max THEN 190
150 LET n3 = n1 + n2
160 LET n1 = n2
170 LET n2 = n3
180 GOTO 140
190 END

RUN
145 PRINT n1

LIST
100 REM Program to print the Fibonacci sequence
110 LET max = 10000
120 LET n1 = 0
130 LET n2 = 1
140 IF n1 > max THEN 190
145 PRINT n1
150 LET n3 = n1 + n2
160 LET n1 = n2
170 LET n2 = n3
180 GOTO 140
190 END

RUN
```



实现一个简化Basic语言的
编辑器和解释器

Modified from Stanford CS106B

Project 3: Minimal Basic

- 目的

- 接触和掌握简单的语法分析和词法分析
- 练习表达式树的使用和表达式求值
- 练习模拟程序执行

- 涉及内容

- 类的设计、继承和多态的使用
- 动态内存管理
- 指针和表达式树的使用
- 程序状态维护和解释执行

- 拓展功能选做

- 支持右结合运算符
- 支持字符串
- 支持函数定义和调用
- 支持外部C函数调用
- 设断点
- ...

第四学期课程（键值存储系统）

- 基于 Log-Structured Merge Tree (LSM)
- LSM Tree是Google开源项目LevelDB和Facebook开源项目RocksDB的核心数据结构
- 是近年来存储学术会议中的热门话题



LSM Tree 键值存储系统

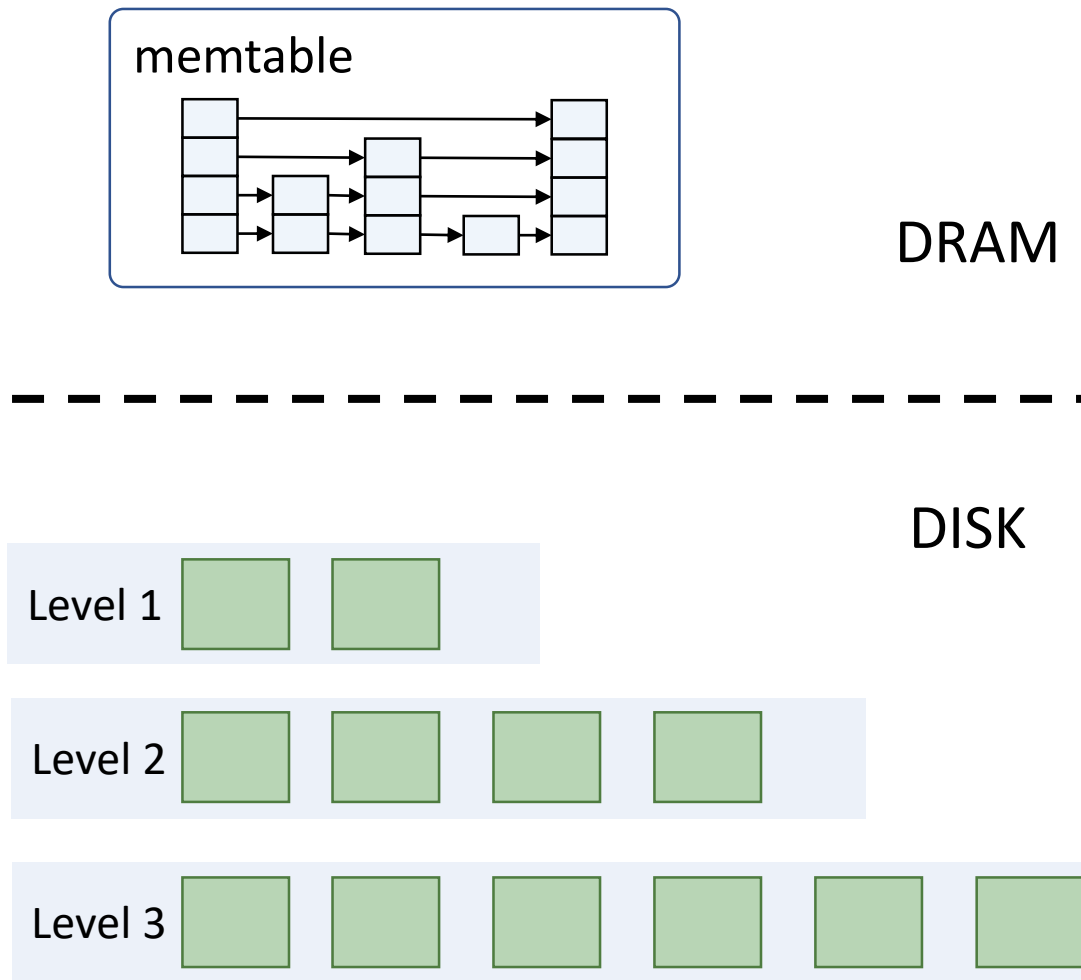
基本操作

- PUT(K,V): 设置键K的值为V
- GET(K): 获取键K的值
- SCAN(K1, K2): 获取键在K1~K2之间的所有键值对
 - 使用迭代器
- DELETE(K): 删除键K和其值

LSM Tree 键值存储系统

基本结构：内存和磁盘两部分

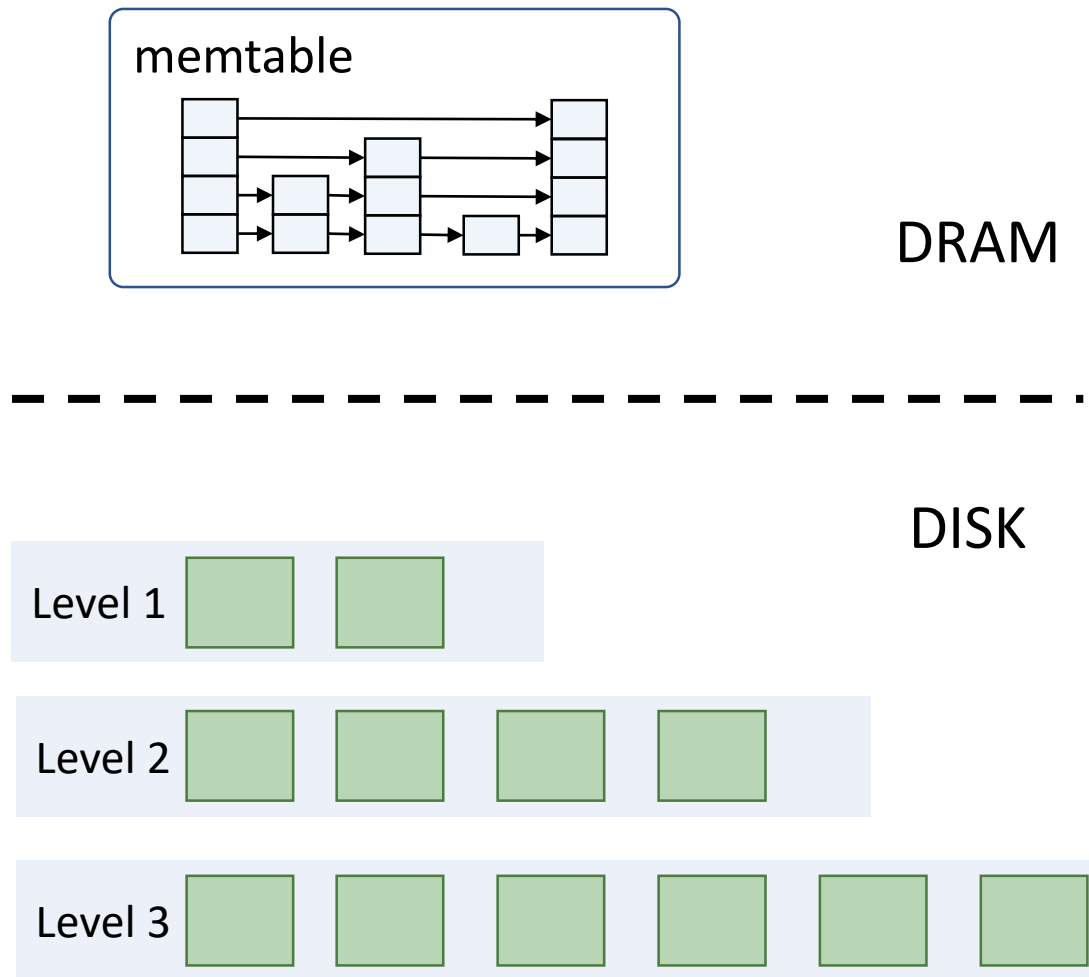
- 内存数据结构：memtable
 - 常用的是跳表（skip-list），也可以选用平衡二叉树等
 - 新写入的数据均被保存在memtable中
- 磁盘中分层保存持久化数据，每层有多个文件（sstable）
 - 每个文件中保存的key是有序的
 - 同一层中文件保存的key区间不相交
 - 越下层文件数量越多，数量比例是预设的



LSM Tree 键值存储系统

合并操作 (compaction)

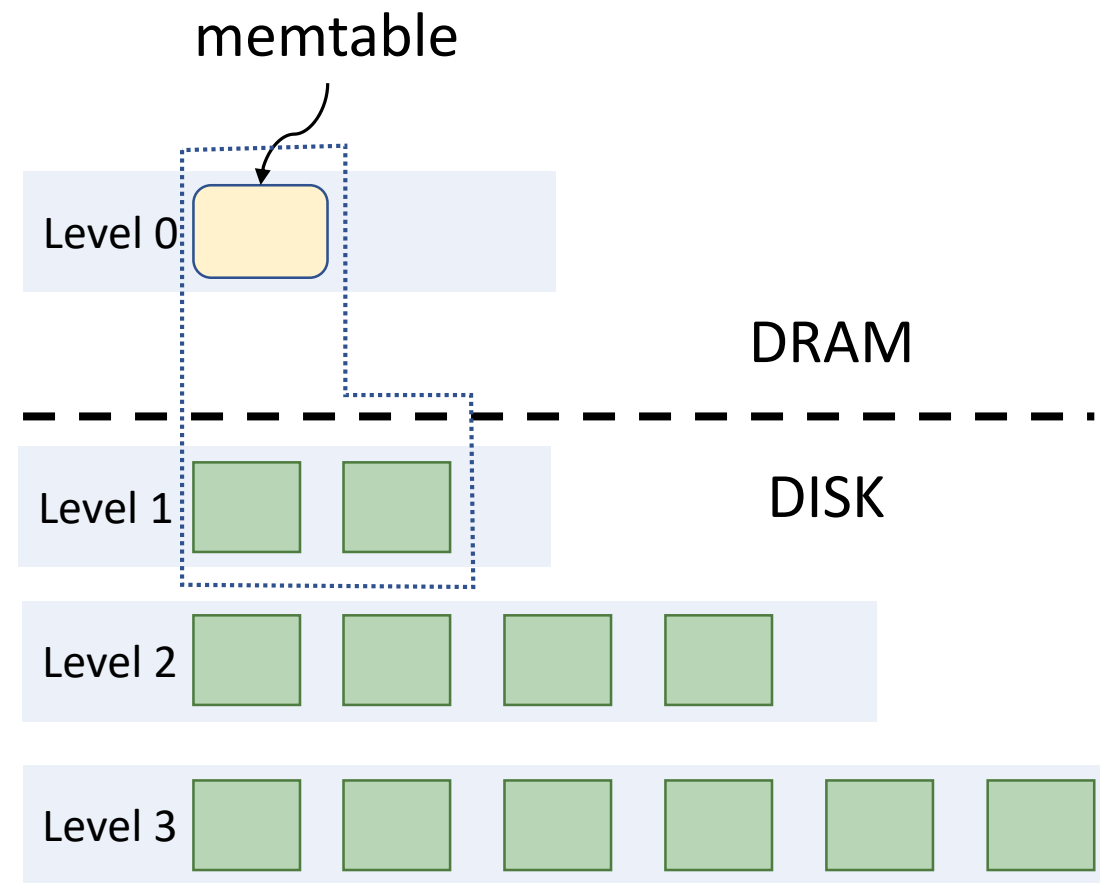
- 当memtable中数据量达到阈值，需将其数据保存到磁盘中
 - 将DRAM中的memtable视为Level 0



LSM Tree 键值存储系统

合并操作 (compaction)

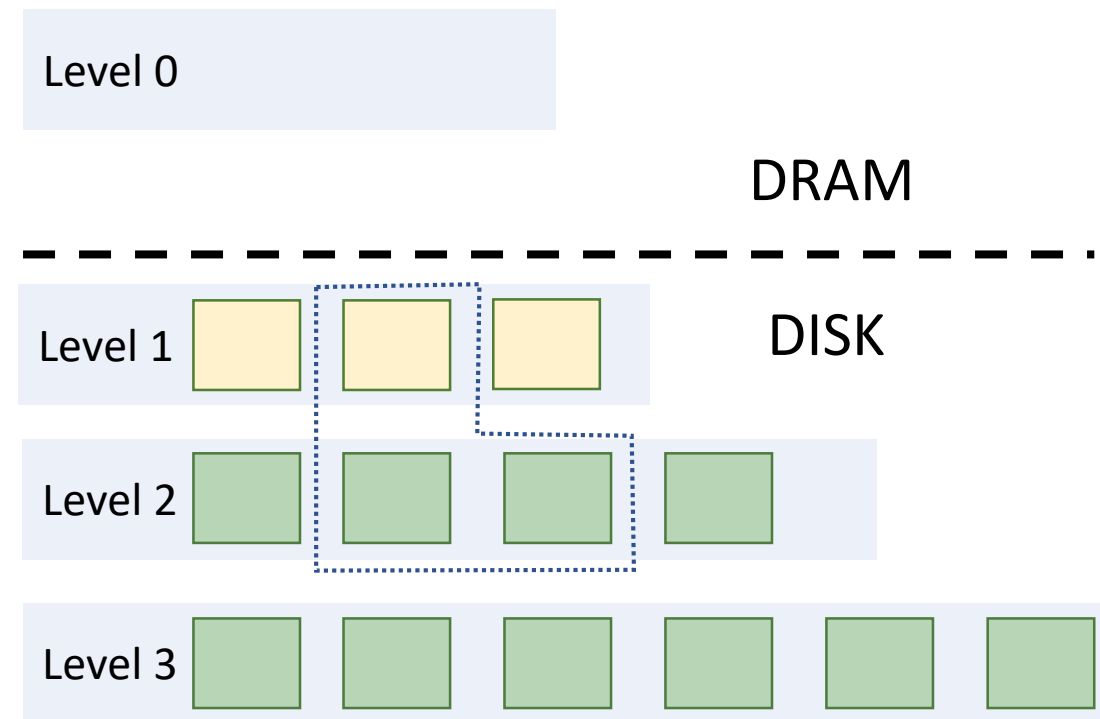
- 当memtable中数据量达到阈值，需将其数据保存到磁盘中
 - 将DRAM中的memtable视为Level 0
 - 将Level 0中的文件与下一层中的文件进行归并排序，结果写入到下一层中



LSM Tree 键值存储系统

合并操作 (compaction)

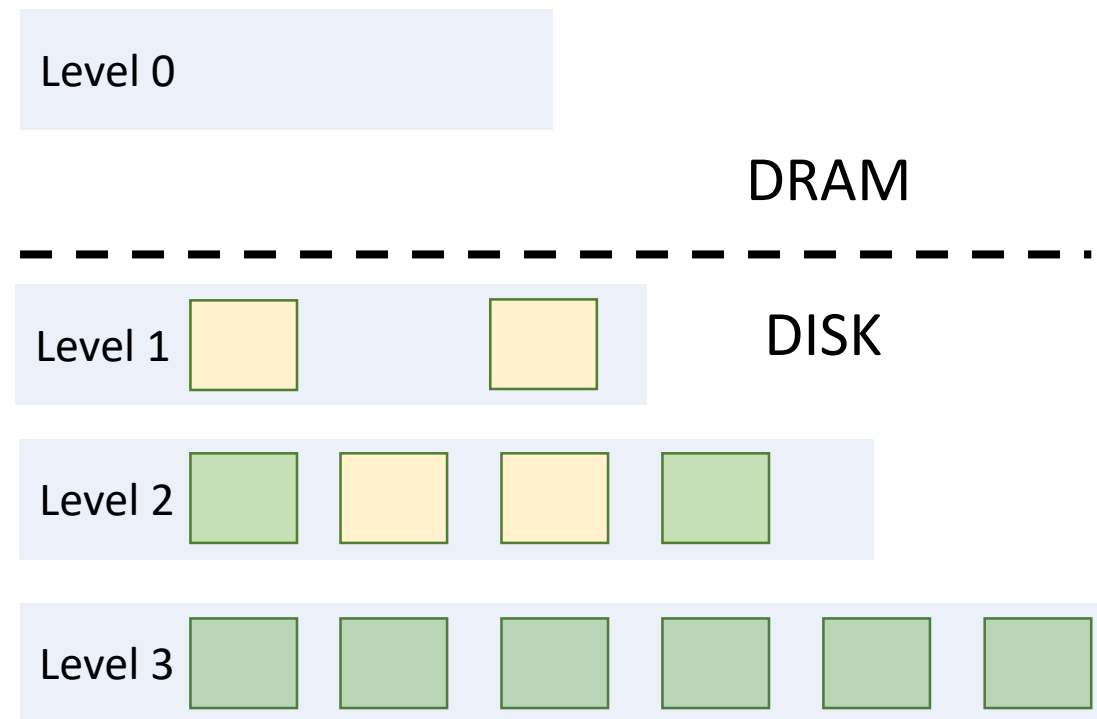
- 当memtable中数据量达到阈值，需将其数据保存到磁盘中
 - 将DRAM中的memtable视为Level 0
 - 将Level 0中的文件与下一层中的文件进行归并排序，结果写入到下一层中
 - 若下一层文件数超出限定，则将其中一个文件继续往更下层合并



LSM Tree 键值存储系统

合并操作 (compaction)

- 当memtable中数据量达到阈值，需将其数据保存到磁盘中
 - 将DRAM中的memtable视为Level 0
 - 将Level 0中的文件与下一层中的文件进行归并排序，结果写入到下一层中
 - 若下一层文件数超出限定，则将其中一个文件继续往更下层合并



LSM Tree 键值存储系统

性能测试和瓶颈分析

- 掌握软件测试方法和性能瓶颈分析
- 熟悉常用的KV测试集
 - 如 Yahoo! Cloud Serving Benchmark (YCSB)

LSM Tree 键值存储系统

可选的优化方向和方法

- 方向
 - 提升合并速度
 - 提升读写性能
 - 减少写放大
 - 提升可靠性
- 方法
 - 使用布隆过滤器 (Bloom Filter) : 快速判断指定Key不在某个sstable中
 - 使用两个memtable: 当一个在合并时, 另外一个可以处理新的写操作
 - 增加 write-ahead-log: 保证写到memtable的数据不会丢失
 - ...

问题

- 如何保证程序的质量
- 可能的解决方法
 - 使用华为开发云
 - 如何解决开发环境?

离散数学 (第三学期)

1 命题逻辑

- 1.1. 什么是命题逻辑
- 1.2. 如何将程序转化为命题逻辑
- 1.3. 利用SAT Solver证明程序的正确性

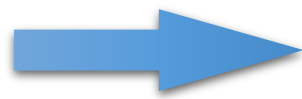


表达式过于复杂

2 一阶逻辑

- 2.1. 什么是一阶逻辑
- 2.2. 将程序转化为一阶逻辑
- 2.3. 利用SMT Solver证明程序的正确性

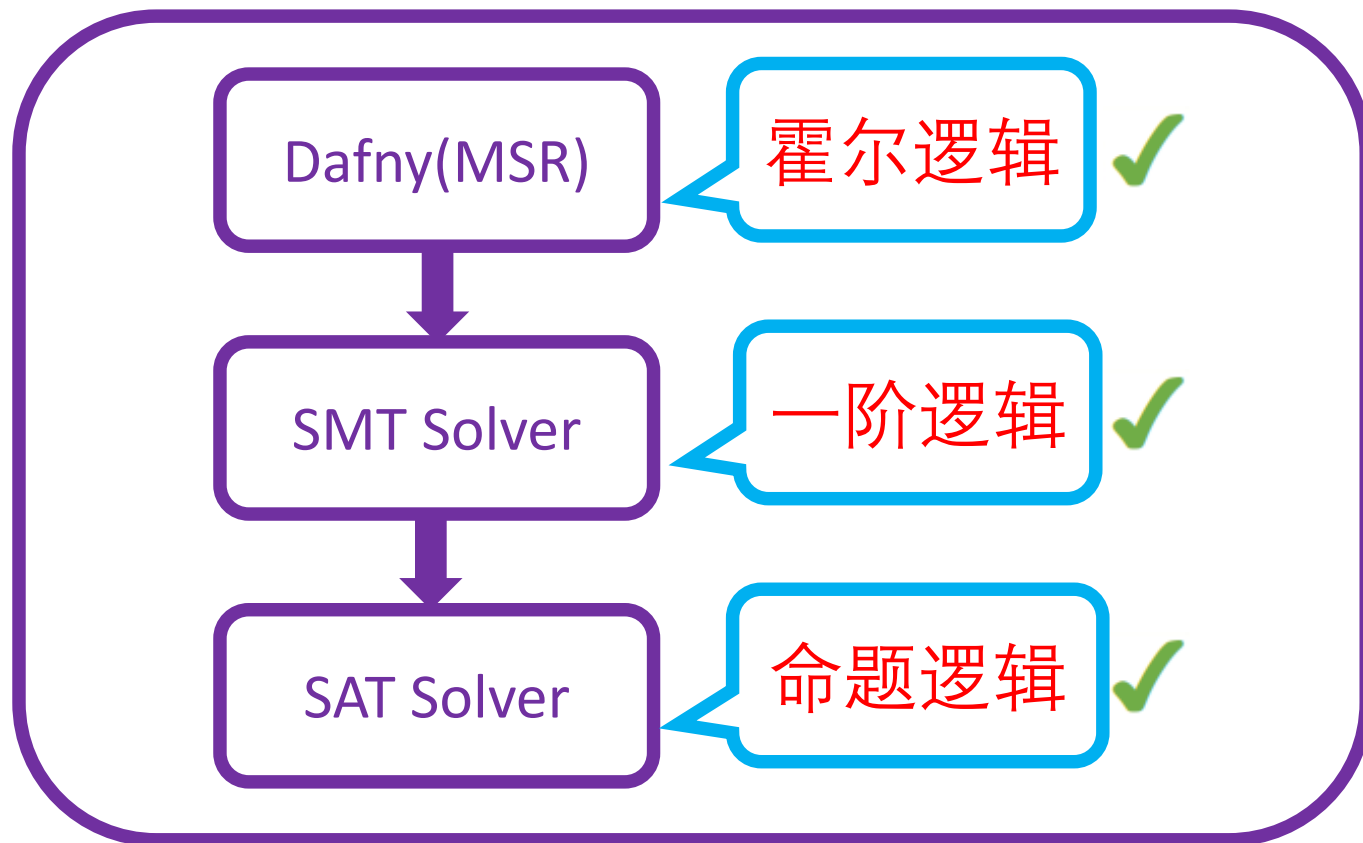
无法证明嵌有循环的程序



3 霍尔逻辑

- 3.1. 公理系统
- 3.2. 霍尔逻辑
- 3.3. 利用霍尔逻辑自动化验证 (Dafny)

离散数学



程序形式化验证

离散数学

```
int BinarySearch(int a[], int len, int value)
{
  int low=0, high=len;
  while(low < high) {
    int mid = (low + high) / 2;
    if(a[mid] < value)
      low := mid + 1;
    else if(value < a[mid])
      high := mid;
    else
      return mid;
  }
  return -1;
}
```



```
method BinarySearch(a: array<int>, value: int) returns (index: int)
  requires 0 <= a.Length && sorted(a)
  ensures 0 <= index ==> index < a.Length && a[index] == value
  ensures index < 0 ==> forall k :: 0<=k<a.Length ==> a[k] != value
{
  var low, high := 0, a.Length;
  while low < high
    invariant 0 <= low <= high <= a.Length
    invariant forall i ::
      0<=i<a.Length && !(low <= i < high) ==> a[i] != value
  {
    var mid := (low + high) / 2;
    if a[mid] < value
      low := mid + 1;
    else if value < a[mid]
      high := mid;
    else
      return mid;
  }
  return -1;
}
```

Dafny 2.3.0.10506

Dafny program verifier finished with 3 verified, 0 errors
Program compiled successfully

谢谢