

## 原 CS61B sp2018笔记 | Efficient Programming

2018年10月24日 18:20:37 隐秀\_ 阅读数: 13 标签: Java 数据结构 更多

# Efficient Programming

"An engineer will do for a dime what any fool will do for a dollar" – Paul Hilfinger

编程的效率取决于以下两方面因素:

### 1. 编程时的消耗 (programming cost)

- 你花费多长时间编写好程序?
- 你的代码是否易读、是否易于更改、易于维护?

### 2. 运行时的消耗 (Execution cost)

- 你的程序将会执行多长时间
- 你的程序将会占用多大内存空间

## Programming cost

我们首先来了解一组概念。

**API** (Application Programming Interface) 是一些预先定义的方法, 每个方法都有简短的描述。

**ADT** (Abstract Data Structures) 是具有类似行为的高级数据类型。

比如, 在 [Proj1a](#) 中, Deque 有具体的相似的行为 (addFirst, addLast 等), 而且我们使用的是它的实现类 [ArrayDeque](#) 和 [LinkedListDeque](#) 所以它 ADT。

一些 ADT 是 其他 ADT 的特殊情况。比如, [Stacks](#) 和 [Queues](#) 就定义了更具体的行为, 而它们仍然是 ADT。

接下来做一个小练习。

使用 [Linked List](#) 实现一个 [Stack](#) 类, 并实现 `push(Item x)` 方法, 确保这个类支持泛型。

下面给出三种常见的实现方法:

```
1 public class ExtensionStack<Item> extends LinkedList<Item> {
2     public void push(Item x) {
3         add(x);
4     }
5 }
```

这种方法通过 [继承](#) 实现, 简单的调用了 [LinkedList<Item>](#) 中的方法。

```
1 public class DelegationStack<Item> {
2     private LinkedList<Item> L = new LinkedList<Item>();
3     public void push(Item x) {
4         L.add(x);
5     }
6 }
```

这种方法通过 [委托](#) (Delegation) 实现, 它在类的内部创造了一个 [Linked List](#) 的对象, 然后调用它的方法完成目标。

```
1 public class StackAdapter<Item> {
2     private List L;
3     public StackAdapter(List<Item> worker) {
```

```

6 |
7 |     public void push(Item x) {
8 |         L.add(x);
9 |     }
10| }

```

这种方法和上一种类似，只不过它可以支持 `List` 接口的任意实现类。

### Delegation vs Extension

通过刚才的练习，我们发现 `Delegation` 和 `Extension` 似乎可以相互转换，但实际上它们之间有一些值得关注的不同之处。

`Extension` 使用前必须清楚父类中发生了什么，换句话说，你必须知道父类的方法是怎样实现的。而使用 `Delegation` 时，你不需要把当前类认为是：的版本，父类的对象只是一个工具而已。

## Asymptotic

### 1. Introduction to Asymptotic Analysis

我们先解决一个问题：如何确定一个数组中含有两个相同的数？

似乎有两种解决方案：

- Silly Algorithm: 每两个数字 之间进行比较，如果相等则返回 true
- Better Algorithm: 好好利用数组的 排序 算法

下面我们根据上述两个思路依次实现

```

1 | //Silly Duplicate: compare everything
2 | public static boolean dup1(int[] A) {
3 |     for (int i = 0; i < A.length; i += 1) {
4 |         for (int j = i + 1; j < A.length; j += 1) {
5 |             if (A[i] == A[j]) {
6 |                 return true;
7 |             }
8 |         }
9 |     }
10|     return false;
11| }
12|
13| //Better Duplicate: compare only neighbors
14| public static boolean dup2(int[] A) {
15|     for (int i = 0; i < A.length - 1; i += 1) {
16|         if (A[i] == A[i + 1]) {
17|             return true;
18|         }
19|     }
20|     return false;
21| }

```

比较两种方法的运行速度，我们有以下几种方法：

- 使用客户端的程序测量运行速度

比较低级的方法是用 `秒表` 测量；如果使用 Unix 系统的话，可以使用 `time` 命令测量；或者可以使用 Princeton Standard 的库，里面有一个 `stopwatch` 测量。

结果表明，当我们输入的数组越大时，`dup1` 将花费更长的时间执行程序，而 `dup2` 的运行时间几乎没有什么变化。

**优点：** 简单方便

**缺点：** 可能会耗费很多时间测试，而且在不同的操作系统、编译器上运行结果也会不同。

- 计算程序执行过程中各个操作的数量

下面是两种方法的统计结果

dup1

operation	symbolic count	count, N=10000
i = 0	1	1
j = i + 1	1 to N	1 to 10000
less than (<)	2 to $(N^2 + 3N + 2)/2$	2 to 50,015,001
increment (+=1)	0 to $(N^2 + N)/2$	0 to 50,005,000
equals (==)	1 to $(N^2 - N)/2$	1 to 49,995,000
array accesses	2 to $N^2 - N$	2 to 99,990,000

dup2

operation	symbolic count	count, N=10000
i = 0	1	1
less than (<)	0 to N	0 to 10000
increment (+=1)	0 to N - 1	0 to 9999
equals (==)	1 to N - 1	1 to 9999
array accesses	2 to 2N - 2	2 to 19998

显然，完成同样的目标 dup2 进行了更少的操作，所以 dup2 方法更佳。

优点： 不受计算机构造的干扰，并且形成了一定的数学模型

缺点： 计算的过程太复杂，而且没有显示程序运行时间

并且，通过上述方法我们可以总结出计算算法时间复杂度的一般步骤：

- 只考虑最差的情况（即操作次数的幂最大）
- 在众多操作中选择一个有代表性的操作

Good choice: **increment** , or **less than** or **equals** or **array accesses**

Bad choice: **ssignment of j**= i + 1, or i = 0

- 忽略幂数较低的部分
- 忽略常数

根据以上四个步骤可以得到 dup2 的时间复杂度是 **N**，dup1 的时间复杂度是 **N^2**

下面做一个练习熟悉简化的过程

**Checkpoint:** What's the shape/orders of growth for the following 5 functions?

function	order of growth
$N^3 + 3N^4$	
$1/N + N^3$	
$1/N + 5$	
$Ne^N + N$	
$40\sin(N) + 4N^2$	

**Answer:**

order of growth
$N^4$
$N^3$
1
$Ne^N$
$N^2$

（原创笔记）加州伯克利大学CS61b数据结构（Java描述）一：对象

OOP(object-oriented programming); object: a repository of data; class: type of object. me...

想对作者说点什么？

我来说两句

**CS61B Homework5作业回顾（附代码）**

👁 408

HW5一开始做不出来的同学可以继续先看Lecture， Encapsulated List那一节老师会回... 来自：[everest115的博客](#)

下载

**Efficient R Programming**

06-20

Efficient R Programming English | ISBN: 1491950781 | 2016 Become a more productive programmer with Efficient R Progr...

下载

**Efficient C++ Performance Programming Techniques(提高c++性能的编程技术)**

12-05

我很厚道的,只要一个资源分..有中英文版的,中文有pdf和png版的... 还是蛮好的一本书,可以看看



**免费开源CRM客户关系管理软件系统**  
百度广告

下载

**Practical and Efficient SAS Programming The Insider's Guide azw3**

01-20

Practical and Efficient SAS Programming The Insider's Guide 英文azw3 本资源转载自网络，如有侵权，请联系上传者或csdn删除 查看...

**CS61B Homework6作业回顾（附代码）**

👁 92

本次作业重点是hascode和compress function， 评价一个compress function好坏的标... 来自：[everest115的博客](#)

**CS61BHomework3作业回顾（附代码）**

👁 127

Part 1.smoosh()函数，目的是将数组中出现的所有数字产生重复的部分删除，例如11... 来自：[everest115的博客](#)

下载

**Efficient C Programming Techniques.pdf**

05-14

Efficient C Programming Techniques.pdf Efficient C Programming Techniques.pdf

**SNIPPER 算法笔记**

👁 624

论文：SNIPER: Efficient Multi-Scale Training 论文链接：<https://arxiv.org/abs/1805.0...> 来自：[AI之路](#)

市场揭秘：从价值投资到快进快出北京老股王只用一招

宏福 · 熾燚

特征选择方法学习**笔记**（二）

之前看过了mRMR的方法了，从特征与目标的最大相似度和特征与特征间的最小冗余... 来自： [zhuchengzhang...](#) 1804

相关热词

cs61b cs61b笔记 伯克利cs61b CS61b怎么用 cs61b难不难

下载

**Efficient programming techniques for ARM.**

08-08

Efficient programming techniques for ARM.

 **AI之路**  

关注

 147篇文章

 **僚机中的战斗机**  

关注

 3篇文章

 **vincent\_chan7**  

关注

 11篇文章

下载

**Efficient C++ Programming**

02-23

Stanley B. Lippman写的一篇文章，可以认真阅读和学习一下

下载

**efficient database programming with abap**

04-02

abap 的数据库编程, SAP ABAP 开发的核心部分

下载

**final review for berkeley cs61b**

07-21

关于伯克利公开课cs61b的期末复习文档

股市第三次机会来了！不看就亏大了！

唐煌投资 · 熾燚

CondenseNet**算法笔记**

论文： CondenseNet: An Efficient DenseNet using Learned Group Convolutions 链接... 来自： [AI之路](#) 2515

论文**笔记**——CFENet

会议： CVPR 2018 标题： 《CFENet: An Accurate and Efficient Single-Shot Object D... 来自： [XJTU\\_Bugdragon](#) 490

下载

**Efficient MIDP Programming**

07-16


Efficient MIDP Programming sun white book

MobileNets **论文笔记**

论文地址： MobileNets: Efficient Convolutional Neural Networks for Mobile Vision A... 来自： [Jesse\\_Mx的博客](#) 1.3万

伯克利计算机低年级核心课程之CS61A-SICP

<http://bbs.eol.cn/forum.php?mod=viewthread&extra=page%3D1&tid=2391800> 给想... 来自： [fengjiexyb的专栏](#) 8775



**硅微粉哪家好？晶盛源值得信赖**  
百度广告

伯克利大学数据结构课程

<http://datastructur.es/sp17/> CS61B-Data Structures UC berkeley 来自： [yaoxiaofeng\\_000...](#) 104

哈佛大学CS50课程**笔记**（网络编程）

0.HTML不是一种编程语言，他不是告诉计算机做什么事情而是一种标记语言告诉计... 来自： [天生愚钝，徐徐...](#) 140

文献阅读**笔记**——road segmentation（一）

一.文献名字和作者 A Fast Forest Road Segmentation for Real-time Robot Self-navig... 来自： [iseehz的博客](#) 472

下载

**Efficient C++ Performance Programming Techniques.rar**

02-17

Efficient C++ Performance Programming Techniques.rar

C++实现控制台版2048（内附彩蛋）

👁 124

前言

之前做过一个JavaScript版本的2048游戏，最近在学习C++，昨天晚上突然...

来自： 隐秀\_

北京股王8年不亏铁律“1272”曝光，震惊众人！

陕西信息科技 · 熯燚

CS61B Homework7作业回顾(附代码)

👁 36

发现编程真的是很强调逻辑。首先Track down the whole tree, split 3-key node into 2 ...

来自： everest115的博客

CS61B Homework9作业回顾（附代码）

👁 57

本次作业主要学会运用Disjoin Sets，DisjointSets的class作业已经写好给出。class中...

来自： everest115的博客

下载 Efficient C++: Performance Programming Techniques

11-27

"这本超薄小书聚焦于高性能C++应用程序开发。两位作者都是IBM软件专家，都工作于对性能要求极高的系统构建领域，本书是他们的经验之谈。"--荣毅(《C++的四书五经》)

下载 Structured Parallel Programming - Patterns for Efficient

06-01

Structured Parallel Programming offers the simplest way for developers to learn patterns for high-performance parallel ...

cs61a课程总结--lecture7 递归（和一种数据结构）

👁 520

数据结构 tuple 是一种至类型

来自： vczhfan的专栏



招聘海归硕士

百度广告

下载 Efficient C++ Performance Programming Techniques

04-03

经典C++进阶书籍 Efficient C++ Performance Programming Techniques

下载 Real-Time C++ (2nd).pdf 2015第2版

12-04

Real-Time C++ Efficient Object-Oriented and Template Microcontroller Programming

下载 Data Structures Lecture Notes (UCB CS61b)

10-20

Data Structures Lecture Notes (UCB CS61b) Data Structures Lecture Notes (UCB CS61b)

《Lua程序设计》(Programming in Lua)阅读笔记

👁 4096

官网： <http://www.lua.org/> 社区： <http://lua-users.org/> 论坛： <http://www.luaer.cn/> ...

来自： 鸟窝

下载 Real-Time C++ Efficient Object-Oriented and Template Microcontroller Programming

08-23

Real-Time C++ Efficient Object-Oriented and Template Microcontroller Programming

老中医说：饭后用一物，体重不过百！北京人必看！

尹承熙 · 熯燚

Efficient c++ 笔记（二）

👁 607

第二章 构造函数和析构函数 2.1 继承 我们应该在不要用到继承的地方避免使用继承...

来自： Briver Song的专栏

Efficient c++ 笔记（三）

👁 520

第三章 虚函数 如果类X定义了一个虚函数或是它派生于这样的类，那么就会由编...

来自： Briver Song的专栏

Efficient C++ 笔记（一）

👁 553

第一章 跟踪范例 在运行跟踪范例Trace的过程中，我们需要根据设置的一个是否需要...

来自： Briver Song的专栏

CS61A 系列课程笔记（一）

👁 1338

嗯 今天刚看第二周的课程，大量的 reading 材料我是真的要崩溃了，全英。。。我...

来自： 柠檬黄先生的博客

下载 论文Efficient Graph- Based Image Segmentation的实现代码，c/c++版本的，支持jpg格式输入输出

07-24

该代码是论文Efficient Graph- Based Image Segmentation的实现代码，c/c++版本的，主要代码来自于该论文的作者pdf的实现代码，在此基础上加入了格式转换的接口使得可以支持j...

老中医说：饭后用一物，体重不过百！北京人必看！

尹承熙 · 熾燚

学习对称形式的量化（Learning Symmetric Quantization）

👁 377

"SYQ: Learning Symmetric Quantization For Efficient Deep Neural Networks&q...来自：AI爱好者

【CVPR2018】论文整理

👁 129

【CVPR2018】论文整理 转载自：https://blog.csdn.net/leviopku/article/details/807...来自：浪小花的博客

LSD-SLAM笔记之一致性约束

👁 903

LSD-SLAM 一致性约束 Sim3求解来自：kokerf的博客

下载

Efficient Graph-Based Image Segmentation

09-03

Efficient Graph-Based Image Segmentation论文的实现代码，以及给了一个测试样例，在Linux下运行通过

如何用matlab画出有效市场边缘

👁 1811

本文介绍如何利用matlab画出两个资产的有效边缘（efficient frontier）假设两项资...来自：Reach Higher



什么是核心期刊 核心期刊有哪些

百度广告

ShuffleNet V2 论文阅读笔记

👁 1565

ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design 阅读笔记 ...来自：Chris\_zhangrx的...

下载

ACE 论文IPC SAP

01-27

IPC SAP C++ Wrappers for Efficient, Portable, and Flexible Network Programming

下载

Efficient C++ Performance Programming Techniques 1999

06-07

Efficient C++ Performance Programming Techniques 1999 pdf (English version)

jQuery选择器优化问题，使用\$('a.b')等，IDE会提示这是低效的用法的解决

👁 1079

当在IDE中这样使用选择器时，会出现这是低效的用法的提示：Checks that jQue...来自：Chengxuyuan\_J...

下载

Quantization and Training of Neural Networks

02-11

Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference



招聘海归硕士

百度广告

ONOS源码笔记--前提

👁 1494

常用服务 @Reference(cardinality = ReferenceCardinality.MANDATORY\_UNARY) prot...来自：fstudio



隐秀\_

关注

原创	粉丝	喜欢	评论
26	22	24	13

等级：

博客 3

访问：3万+

积分：503

排名：11万+

勋章：



1024



### 十大笔记本排名



about

cs专业一名挣扎的后端  
热爱交友

GitHub: [github.com/seriouszyx](https://github.com/seriouszyx)  
cnblog: [cnblogs.com/henuzyx](https://cnblogs.com/henuzyx)  
独立博客: [seriouszyx.github.io](https://seriouszyx.github.io)

热门文章

[最新超详细VMware虚拟机下载与安装](#)  
阅读量: 19735

[最新超详细虚拟机VMware安装Kali Linux](#)  
阅读量: 5596

[上机考试作弊，也不是不可以呢](#)  
阅读量: 2642

[hexo搭建个人独立博客 | Next主题深度美化方案](#)  
阅读量: 660

[JSONArray.fromObject不执行且不报错问题的解决](#)  
阅读量: 431

最新文章

[Tomcat 引入jar包导致的  
NoClassDefFoundError 报错的问题解决](#)

[Spring IoC容器浅析及简单实现](#)

[Linux 基本操作总结（后端必备）](#)

[Java生成解析二维码](#)

[Ubuntu 安装 IntelliJ idea \(ultimate\)](#)

博主专栏



CS61B sp2018笔记

阅读量: 514    6 篇

个人分类

C++

4篇

JavaScript

5篇

Linux

4篇

游戏

2篇

Java

13篇

展开

归档

2018年10月

2篇



2018年9月	1篇
2018年8月	4篇
2018年7月	8篇
2018年6月	4篇

[展开](#)

#### 最新评论

##### hexo搭建个人独立博客 | Ne...

qq\_40950957: [reply]qq\_39901385[/reply] 我用的是这个主题 NexT.Gemini, 我...

##### hexo搭建个人独立博客 | Ne...

qq\_39901385: 我知道什么问题了是没清缓存 楼主就是那个主页怎么把一篇文字折叠起来呀 用一个按钮来控制打开这个怎么...

##### hexo搭建个人独立博客 | Ne...

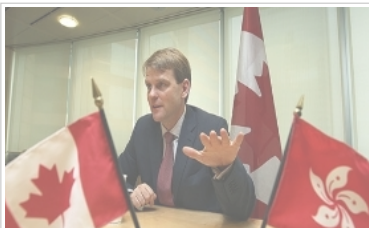
qq\_39901385: 本地能设置next的背景和头像旋转 使用了hexo g -d放在了git上 绑定了域名 但是通过g...

##### hexo搭建个人独立博客 | Ne...

qq\_39901385: 楼主方便加个q吗我有些hexo搭配next的主题优化的问题想问你

##### 最新超详细VMware虚拟机下载与...

qq\_43318544: 不错顶起来, 现在有15了VMware Workstation Pro15虚拟机破解版: [https...](https://blog.csdn.net/qq_40950957/article/details/82215410)



#### 加拿大移民新政



#### 联系我们



扫码联系客服



区块链大本营

QQ客服

kefu@csdn.net

客服论坛

400-660-0108

工作时间 8:00-22:00

[关于我们](#) [招聘](#) [广告服务](#) [网站地图](#)

百度提供站内搜索 京ICP证09002463号

©2018 CSDN版权所有

网络110报警服务 经营性网站备案信息

北京互联网违法和不良信息举报中心

中国互联网举报中心