

真题勘误:

一, 2003 年数据结构第三题:

3. 第一步求出散列值

4	4	3	6	4	1	5	2	1	3	4	7
0	6	4	3	5	1	7	4	7	3	7	9
6	1	0	1	1	1	6	7	0	1	1	1
	2		2	1	1				6	3	1

溢出处理: $H_i = (H_i + i) \% m$, 其中 H_i 表示要插入的元素, i 为 $0, 1, 2, \dots, m$ 为 17。

0	1	2	3	4	5	6	7	8	9	1	1	1	1	1	1	1
										0	1	2	3	4	5	6
3	1	7				4	5	2			4	4	6	1	4	3
4	7	9				0	7	4			5	6	3	1	7	3

二, 2008 操作系统第七题

7. (1) 1 (直接访问) (2) 25 (指针顺序访问) (3) 2 (一次索引块, 一次读第 25 块)

三, 2017 数据结构第一题

C 改为 D

四, 2018 年数据结构第一题

5 改为 s

五, 2013 年操作系统期末 A 第八大题

答: 11: 10 12: 00
 11: 40 12: 20
 11: 00 11: 10
 11: 10 11: 40

六, 2017 年软工第八题答案代码改成

```
Public class Employee {
    Private Double income;
    Public Double getIncome(){
        Return income;
    }
}

Public class Tax {
    Private Double taxrate;
    Private Employee employee;
    Public Double getTaxrate(){
        Return taxrate;
    }
    Public Double getTax() {
        employee = new Employee();
        Return employee.getIncome()*taxrate;
    }
}
```

七，2014 年操作系统 46 题答案改为

J1 装入主存时间:11:10____， 结束时间:____12:00__；

八，2003 年数据结构第五题代码改为

```
Type def struct BiNode{
    Int Element;
    Struct BiNode Leftchild,Rightchild;
}BiNode *BiNode;
Int getDepth(BiTree *T){
    Int LD,RD;
    If(T == Null) return 0;
    Else if(!T->Leftchild)&&(!T->Rightchild))
    Return 1;
    Else{
        LD = getDepth(T->Leftchild);
        RD = getDepth(T->Rightchild);
        Return ((LD>RD)?LD:RD);
    }
}
```

九，2007 年操作系统第五题改为

1) **FIFO**,缺页次数 **6**, 缺页中断率为 **6/13**

	3	2	1	4	4	5	5	3	4	3	2	1	5
1	1	1	1	1	1	1	1	3	3	3	3	3	5
2	2	2	2	2	2	5	5	5	5	5	5	1	1
3	3	3	3	4	4	4	4	4	4	4	2	2	2
是否缺页				✓		✓		✓			✓	✓	✓

2) **LRU**, 缺页次数 **6**, 缺页中断率为 **6/13**

	3	2	1	4	4	5	5	3	4	3	2	1	5
1	1	1	1	1	1	1	1	3	3	3	3	3	5
2	2	2	2	2	2	5	5	5	5	5	2	2	2
3	3	3	3	4	4	4	4	4	4	4	4	1	1
是否缺页				✓		✓		✓			✓	✓	✓

十，2016 年数据结构第一题

(1)

K		B	M	D	CI	X			I	TM
1		1	2	1	3	5			1	2

(2) 总的比较次数为 16

十一，2013 年选择题第一题选 D

十二，2015 年选择 10 选 C,8 改为 D

十三，2014 年选择 35 改为 A

十四，2016 年软工第四题 (4) 改为约束 (3) (5) 改为系统级需求, (1) 改为业务需求

十五，2016 年操作系统第二题次数改为 250，第四题控制块，程序块，数据块，核心块

十六，2016 年操作系统第九题 (1) 改为

9. $q=1$

R		A	B	C	B	D	B	D	B	D	B	D	B	D	B	D	B	D	D
R																			

十七，2013 备用卷选择题 33 改为 B，35 改为 C，38 改为 C

十八，2017 年操作系统第五题题目有误

5、考虑下面的进程集合：

进程	到达时间	处理时间
A	0	2
B	1	6
C	2	2
D	3	6

如果使用先来先去服务 **FCFS** 调度算法，得到的每个单位时间内的进程执行序列表示为

算法	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
FCFS	A	A	B	B	B	B	B	B	C	C	D	D	D	D	D	D

参照 **FCFS** 调度算法给出的执行序列的写法，试写出如下三个调度算法那的执行序列，即在表格中填入每个单位时间内执行的进程代号。

- (1) 时间片轮转调度算法 (**RR**, $q=1$, 即时间片长度为 1) ; (2 分)
- (2) 时间片轮转调度算法 (**RR**, $q=4$, 即时间片长度为 4) ; (2 分)
- (3) 多级反馈队列算法 (**Fback**, $q=1$, 即各个多级反馈队列的时间片长度均为 1) 。 (2 分)

5. $q=1$

R		A	B	A	C	B	D	C	B	D	B	D	B	D	B	D	D
R																	

$q=4$

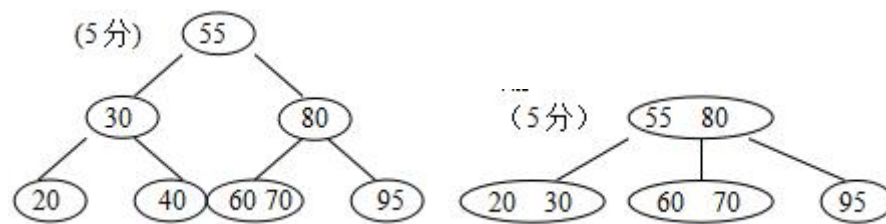
RR		A	A	B	B	B	B	C	C	D	D	D	D	B	B	D	D
----	--	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Feedback

R		A	B	C	D	A	B	C	D	B	D	B	D	B	D	B	D
R																	

十九，2013 年计网最后一题改为 A: 26, B: 28, C: 26, D: 28, 答案均不唯一

二十，2005 年数据结构第二题的第三小题改为：



二十一，2013 年第四十三题改为
有理数类：

```

public class Rational {
    public int i;
    public int j;
    public Rational(){
        i=1;
        j=1;
    };
    public Rational(int I,int J)
    {
        i=I;
        j=J;
    }
    public void setRational(int I,int J)
    {
        i=I;
        j=J;
    }
    public double getI(){return i;}
    public double getJ(){return j;}
    public Rational sum(Rational b){
        Rational su=new Rational();
        su.setRational((int)(i*b.getJ()+j*b.getI()),(int)(b.getJ()*j));
        return su;
    }
    public Rational subtraction(Rational b){
        Rational sub=new Rational();
        sub.setRational((int)(i*b.getJ()-j*b.getI()),(int)(j*b.getJ()));
        return sub;
    }
}
  
```

```

    }
    public Rational multiplication(Rational b){
        Rational mul=new Rational();
        mul.setRational((int)(i*b.getI()),(int)(j*b.getJ()));
        return mul;
    }
    public Rational division(Rational b){
        Rational div=new Rational();
        div.setRational((int)(i*b.getJ()),(int)(j*b.getI()));
        return div;
    }
    public String toString()
    {
        return i+"/"+j;
    }
}

```

有理数类的测试:

```

public class RationalTest {
    public static void main(String[]args)
    {
        Scanner in=new Scanner(System.in);
        System.out.println("please enter two  number");
        Rational a=new Rational(in.nextInt(),in.nextInt());
        System.out.println("please enter two number");
        Rational b=new Rational(in.nextInt(),in.nextInt());
        System.out.println("Rational a:"+a);
        System.out.println("Rational b:"+b);

        System.out.println("a+b:"+ a.sum(b).toString());

        System.out.println("a-b:"+a.subtraction(b).toString());

        System.out.println("a*b:"+a.multiplication(b).toString());

        System.out.println("a/b:"+a.division(b).toString());

    }
}

```

2015 年操作系统大题第一题改为

1.直接块容量= $12 \times 1024\text{B}/1024=12\text{KB}$

一次间接容量=256*1024B/1024=256KB
 二次间接容量=256*256*1024B/1024=65536KB
 三次间接容量=256*256*256*1024B/1024=1677216KB
 28MB=28672KB
 28672-12-256=28404KB
 28404*1024B/1024=28404 块
 所以占二级 28404 块，一级 256 块，直接 12 块

2005 年真题数据结构部分解答 2 扩充 4 叉树，需要将图中每行按照大小顺序排

2006 年数据结构部分解答 (1) a 下面字母依次添加 1 1 2 1 3 5 1 2

(3) 原图中重复的 1 改为 3，答案中 53 改为 52

2008 年解答 (3) 第一个图数字改为 70

2018 年数据结构部分大题最后一题与 2008 操作相同

2017 年软工第九题去重复的题严谨一点暂且可以改为

```

    }
    private Double getTotalSumExcludeCanaelAmount(List amounts){

        double totalToPay = 0;
        Iterator amountsIterator = amounts.iterator();
        while (amountsIterator.hasNext()){
            Amount amount = (Amount)amountsIterator.next();
            //需要排除
            if (!amount.getIsToCancel()) {
                totalToPay += getpay(amount);
            }
            //如果不需要去掉if
        }
        return new Double(totalToPay);
    }
    private double getpay(Amount amount){
        String cancelstatuses = "aassdd";
        if (!cancelstatuses.contains(amount.getStatus())){
            return amount.dubleValue();
        }
        return 0.0;
    }
}
  
```

2017 年软工第九题

九、面向对象

Public interface process() {}

Public class addCourse() implements process();

Public class removeCourse() implements process();

2016 年软工第六题去掉内聚太高应为内聚不够

2015 年操作系统第二题 opt 算法最后两次替换改为 712, 712

2014 年软工 45 题

用工厂如下:

```
public class Department {
    private List<Employee> employeeList;
    private simpleFactory factory;
    public Department(simpleFactory factory){
        this.factory = factory;
    }
    public Employee addEmployee(int employeeType,String name){
        Employee emp = factory.createEmployee(employeeType,name);
        emp.setDepartment(this);
        employeeList.add(emp);
        update();
        return emp;
    }
}

public class simpleFactory{

    public Employee createEmployee(int employeeType,String name){
        Employee emp;
        switch (employeeType){

            case EMPLOYEE.SALARIED: emp = new SalariedEmployee(name);break;
            case EMPLOYEE-HOUR: emp= new HourlyEmployee(name); break;
            case EMPLOYEE.COMMISSION:emp=new ComrmissionedEmployee(name); break;
            default: emp= new Employee(name);

        }
        return emp;
    }
}
```

2015 年软工第八大题方法名统一改为 compare (……)

2013 操作系统 AB 卷中管程的相关题目去 ppt 中找

大题四改为

Clock 算法

	0	1	7	0	1	2	0	1	2	3	2	7
页框 0	0*	0*	->0*	->0*	->0*	2*	2*	->2*	->2*	3*	3*	3*
页框 1	->	1*	1*	1*	1*	->1	0*	0*	0*	->0	2*	2*
页框 2		->	7*	7*	7*	7	->7	1*	1*	1	->1	->7*
缺页标记	F	F	F			F	F	F		F	F	F

缺页次数为__9__次

操作系统样卷中选择题 17 改为 B

大题四改为

Clock 算法：缺页次数为__9__次。

	4	3	2	1	4	3	5	4	3	2	1	5
页框 0	4*	4*	→4*	1*	1*	→1*	5*	5*	5*	5	→5	→5*
页框 1	→	3*	3*	→3	4*	4*	→4	→4*	→4*	2*	2*	2*
页框 2		→	2*	2	→2	3*	3	3	3*	→3	1*	1*
缺页标记	F	F	F	F	F	F	F			F	F	

2013 计网期末单选题 14 改为 D，多选题 1 改为 ACD，3 改为 CE，4 改为 ABD，5 改为 AD，6 改为 BC，8 改为 AEF，(AB 卷对应题)

B 卷 15 改为 C

A 卷第六题改为

6. A.192.168.20.0/26 A=60+1 $2^6-2=62$ 192.168.20.0 0 xx xxxx/26
 B.192.168.20.128/28 B=6+1 $2^3-2=6$ 192.168.20.0 1 0 0 1xxx/28
 C.192.168.20.160/28 C=12+1 $2^4-2=14$ 192.168.20.0 1 0 1 xxxx/28
 D.192.168.20.64/26 D=31+1 $2^6-2=62$ 192.168.20.0 1 1 x xxxx/26

B 卷对应题作同样修改，并且 20 改为 0

软工 B 卷第七大题无限不循环数才为无理数,第二题改为:
图也相应的自己作修改

```
public class Borrower{
    Puvlic void borrowBook();
    public void notifyReturnBook();
}
class BachlorBo extends Borrower{
    int MAX_FOR BACHLOR=5;
    private void borrow BookforBachlor(){
        borrowBook();
        NotifyReturnBook();
    }
}
class MasterBo extends Borrower{
    int MAX_FOR BACHLOR=10;
    private void borrow BookforMaster(){
        borrowBook();
        NotifyReturnBook();
    }
}
class TeacherBo extends Borrower{
    int MAX_FOR BACHLOR=20;
    private void borrow BookforTeacher(){
        borrowBook();
        NotifyReturnBook();
    }
}
```

2013 年操作系统 A 卷 258KB 改为 256KB,B 卷一样。