**数据结构**

**其他**

本课程所介绍的一些算法不数据结构，乃是针对实际应用中普遍存在的非随机数据集而设计的；反过来，只要数据集是理想随机的，则大可不必采用。试举三个这样的案例，列出讲义页码，并作简要说明（各不超过两行）。（2019.1期末）

随机算法的复杂度往往会呈现几何级数的形式因而可计算期望成本，试给出四个例子并简单说明。（2020.1期末-回忆版本1）

在本课的学习中，曾出现过许多利用几何分布（算法执行到下一步停止的概率p）进行计算的实例，请举出4处并简要说明之。（2020.1期末-回忆版本2）

**大O记号**

即便f(n) = O(g(n))，也未必2^(f(n)) = O(2^g(n)) （2014期中）

若每一递归实例本身仅需常数时间和空间，则（）函数的渐进时间复杂度等于渐进空间复杂度。（2014期中）

A) 尾递归   B）线性递归 C）二分递归 D）多分支递归

设图灵机在初始状态下，只有读写头所对单元格为'0'，其余均为'#'；此后，连续地执行increase()算法2014次。在此期间，读写头累计移动的次数（就相对误差率而言）最接近于（）。（2014期中）

A) 2000   B）4000 C）8000 D）16000 E）32000

f(n) = O( g(n))，当且仅当 g(n)=Ω(f(n)) 。（2011期中）

若f(n) = O(n^2) 且g(n)=O(n)，则下列结论正确的是：（2011期中）

A. f(n) + g(n) = O(n^2)

B. f(n) / g(n) = O(n^2)

C.g(n)=O(f(n))

D.f(n) \* g(n) = O(n^3)

算法 g(n)的复杂度为 Θ(n)。若算法 f(n)中有 5 条调用 g(n)的指令，则 f(n)的复杂度为：（2011期中）

A．Θ(n) B.O(n)      C. Ω( n)   D. 不确定

**向量、栈**

对有序向量做 Fibonacci 查找，就最坏情况而言，成功查找所需的比较次数与失败查找相等。（2011期中）

RPN 中各操作数的相对次序，与原中缀表达式完全一致。（2011期中）

对不含括号的中缀表达式求值时，操作法栈的容量可以固定为某一常数。（2011期中）

无论有序向量或有序列表，最坏情况下均可在O(logn)时间内完成一次查找。（2011期中）

对于同一有序向量，每次折半查找绝不会慢于顺序查找。（2011期中）

使用binsearch算法版本C在有序向量{1，3，5，.....2013}中查找，目标为独立均匀分布于[0,2014]内的整数。若平均失败查找长度为F，则平均成功长度S应为（）。（2014期中）

A) 1008F/1007 +1

B）1008F/1007 -1

C） 1008(F-1)/1007 +1

D）1008(F+1)/1007 -1

字符串"123XY"中的字符经栈混洗之后，可得到（）个合法的C++变量名（比如"YX321"）（2014期中）

A) 28   B）5   C）6     D）5     E）以上皆非

evaluate()算法的优先级表中，有的空格项对应于表达式不合法或不合常识的情况，比如（）（2014期中）

A) pri[‘\0’][‘)’]   B）pri[‘!’][‘(’]     C）pri[‘)’][‘!’]    D）pri[‘(’][‘\0’]

实际上，evaluate()算法居然可以对非法表达式“(12)3+!4\*+5”进行求值，其返回值为（2014期中）

A）41   B）89   C）365    D）以上皆非

共有几种栈混洗方案，可以使字符序列{‘x’,’o’,’o’,’o’,’x’}的输出保持原样？（2011期中）

A．12     B. 10       C. 6        D. 5

对长度为n =Fib(k)-1的有向序列做 Fibonacci 查找。若各元素的数值等概率独立均匀分布，且平均成功查找长度为 L，则失败平均查找长度为：（2011期中）

A．n(L-1)/(n-1)     B. n(L+1)/(n+1)      C. (n-1)L/n     D. (n+1)L/n

对长度为 Fib(12) – 1 = 143 的有序向量做 Fibonacci 查找，比较操作的次数至多为：（2011期中）

A．12     B. 11      C. 10     D. 9

考察表达式求值算法。算法执行过程中的某时刻，若操作符栈中的括号多达 2010 个，则此时栈的规模（含栈底的’\n’）至多可能多达？试说明理由，并示范性地画出当时栈中的内容。（2011期中）

设整数e独立且均匀地取自[0,25)，现通过调用fibSearch(A，e，0，7),对如下整数向量A[]做查找：（2014期中）

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| k | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| A[k] | 1 | 3 | 5 | 7 | 9 | 17 | 19 |

试分别计算其在失败情况下的平均查找长度，以及总体的平均查找长度。

**二叉树，BST**

**判断**

**AVL**

若 AVL 树插入元素的过程中发生了旋转操作，则树高必不变。（2016.1期末）

将0..2^d-1插入AVL一定高度为d 。（2014期末补充）

在某节点被删除后AVL树的高度即便下降了，这次操作期间也未必做过旋转调整。（2019.1期末）

对规模为n的AVL树做一次插入操作，最坏情况下可能引发Ω(logn) 次局部重构。（2019.1期末）

设在某新节点插⼊AVL树后（尚待平衡化时），最低失衡节点为g。若此时g的左、右孩⼦的平衡因⼦分别为-1，0，则应通过（ ）旋转使之重新恢复平衡。（2014期末）

- A）zig；

- B）zig+zag；

- C）zag+zig；

- D）zag；

- E）不确定

将[1481,1992]区间内的整数逐一插入到空AVL树中，最后该AVL树的高度是：（2012.6期中）

A、7 B、8 C、9 D、10 E、以上都不对

**伸展树splay**

在任何情况下，伸展树总能保持每次操作 *O*(log *n*) 的平均复杂度。（2016.1期末）

即便访问序列不满足局部性（⽐如完全理想的随机），伸展树依然能够保证分摊O(logn)的性能。（2014期末）

将2014个数插入splay，第一次访问经过2013次旋转，则是单调插入的。（2014期末补充）

只有在访问序列具有较强的局部性时，伸展树才能保证分摊O(logn) 的性能。（2019.1期末）

将{0,1,2,.....,2018}插入一棵空的伸展树后若树高为2018 ，则上述词条必是按单调次序插入的。（2019.1期末）

最底局的叶节点一旦被访问（并做过splay调整）之后，伸展树的高度必然随即下降。（2019.1期末）

**红黑树**

若红黑树插入一个元素后，黑高度增加，则双红修正过程中没有拓扑结构变换，只有重染色操作。（2016.1期末）

红黑树的插入或删除操作，都有可能导致Ω(logn) 个节点的颜色反转。（2019.1期末）

在插入操作后若红黑树黑高度增加，则在双红修复过程中仅做过重染色，而无任何结构调整。（2019.1期末）

对红黑树进行插入操作时，进行双红修正，黑高度增加，则\_\_\_\_\_发生重染色，\_\_\_\_\_发生结构调整。（2012.6期中）

（两个空分别可以填入“必然”、“可能”、“必然不”，选项就是这三者的排列组合）

**综合题**-红黑树（2014期末补充）

依次插入[0,N)

1、写出N=9的红黑树（md我把9也插进去了T\_T）

2、写出树高H和N的通项公式（据说习题解析里有）

红黑树结构，如果不显式记录颜色，通过隐式记录应该如何操作？（2020.1期末）

**B树**

将 *N* 个关键码按随机次序插入 B 树，则期望的分裂次数为 *O*(log2 *N*)。（2016.1期末）

我们知道，BTree:solveOverflflow()和BTree:solveUnderflflow()在最坏情况下均需下界(logn)的时间，然⽽在B-树⼈意⾜够⻓的⽣命期内，就分摊意义⽽⾔⼆者都仅需要O(1)时间。（2014期末）

B树的任一非叶节点内，每个关键码都存在直接后继，且必然来自某个叶节点。（2019.1期末）

人类拥有的数字化数据数量，在 2010 年已达到 ZB(270 = 1021) 量级。若每个字节自成一个关键码，用一颗 16 阶 B-树存放，则可能的高度为（ ）（2016.1期末）

A. 10

B. 20

C. 40

D. 80

E. >80

将[23, 1481)区间内的整数组成一个2-3-B树，且根节点只有一个关键码，则最终该B-树的高度至少是（2012.6期中）

A、7 B、8 C、9 D、10 E、以上都不对

考查包含2018个关键码的16阶B-树，约定根节点常驻内存，且在各节点内部采用顸序查找。（2019.1期末）

a) 在单次成功查找的过程中，至多可能需要读多少次磁盘？请列出估算的依据。

b) 在单次成功查找的过程中，至多可能有多少个关键码需要不目标关键码做比较？请列出估算的依据。

选一小题做即可（2020.1期末）

1)2019阶的B树 插入某关键码后树高增加，此时再次删除该关键码后树高一定降低? 给出证明或反例

(2)2019阶的B树 删除某关键码后树高降低，此时再次插入该关键码后树高一定增加? 给出证明或反例

如果每个结点占用 2 个磁盘块因而需要 2 次磁盘访问才能实现读写，那么在一棵有 n 个关键码的 2m 阶 B 树中，每次搜索需要的最大磁盘访问次数是多少？（2010年期末A卷-感觉此卷风格不是邓公）

给定一棵保存有 n 个关键码的 m 阶 B 树。从某一非叶结点中删除一个关键码需要的最大磁盘访问次数是多少？（2010年期末A卷-感觉此卷风格不是邓公）

**BST其他**

在 BST 中删除两个节点（7B3），则无论先删除哪个节点，最终 BST 的拓扑结构均相同。（2016.1期末）

如果元素理想随机，那么对二叉搜索树做平衡化处理，对改进其渐进时间复杂度并没有什么卵用。（2016.1期末）

由同⼀组共n个词条构成的任意两棵BST，经O(logn)次zig或zag旋转之后，必定可以相互转换。（2014期末）

在 kd-search 中，查找区间 R 与任一节点的 4 个孙节点（假设存在）对应区域最多有 2 个相交。（2016.1期末）

带权重的最优PFC编码树不仅未必唯一、拓扑结构未必相同，甚至树高也可能不等。（2019.1期末）

若调用BST::remove(e) 将节点x 从常规BST中删除，则所需的时间为被删除之前x的深度。 （2019.1期末）

由 5 个互异节点构成的不同的 BST 共有（ ）个 （2016.1期末）

A. 24

B. 30

C. 36

D. 42

E. 120

以下数据结构，在插入元素后可能导致 *O*(log *n*) 次局部结构调整的是（ ）（2016.1期末）

A. AVL

B. B-树

C. 红黑树

D. 伸展树

E. 以上皆非

以下数据结构，空间复杂度为线性的是（ ）（2016.1期末）

A. 2d-tree

B. range tree

C. interval tree

D. segment tree

E. 以上皆非

在 BST 中查找 365，以下查找序列中不可能出现的是（ ）（2016.1期末）

A. 912, 204, 911, 265, 344, 380, 365

B. 89, 768, 456, 372, 326, 378, 365

C. 48, 260, 570, 302, 340, 380, 361, 365

D. 726, 521, 201, 328, 384, 319, 365

以下数据结构中，空间复杂度不超过线性的有（ ）。（2014期末）

- A）2d-tree

- B）3d-tree

- C）2D range tree

- D）interval tree

- E）segment tree

- F）priority search tree

若将有根有序的多叉树T所对应的二叉树记作B(T)，则T的（）遍历序列与B(T)的（）遍历序列完全相同。（2014期中）

A）后序....后序   B）后序....中序） C）层序....先序   D）先序....先序   E）以上皆非

在二叉树（）遍历序列中，祖先节点一定位于其后代节点之前。（2014期中）

A）先序   B）中序   C）后序   D）层次  E）以上皆非

在Huffman编码算法中，若每次（超）字符合并时均保证左兄弟不小于右兄弟，则在所生成的编码树的层次遍历序列中，（）必然按其频率的非升次序排列。（2014期中）

A）（仅）叶节点   B）（仅）内部节点  C）所有节点  D）以上皆非

**填空**：在由2014个节点构成的完全二叉树做层次遍历，辅助队列的容量至少应为（ ）；在整个遍历过程中，辅助队列的规模共在（ ）步迭代中处于这一规模。（2014期中）

对以下各搜索树进行删除操作，哪些树可能会经过Ω(logn)次局部调整，其中n为关键码的数量。（2012.6期中）

A、AVL B、伸展树 C、红黑树 D、B-树 E、都不会

**综合题**

在不改变 BST 和 BinNode 定义的前提下（BinNode 仅存储 parent, data, lc,rc），设计算法，使得从节点 *x* 出发，查找值为 *Y* 的节点 *y* 的时间复杂度为 *o*(*d*)， *d* 为节点 *x* 与 *y* 的距离。要求利用树的局部性，复杂度与总树高无关，否则将不能按满分起评。（2016.1期末）

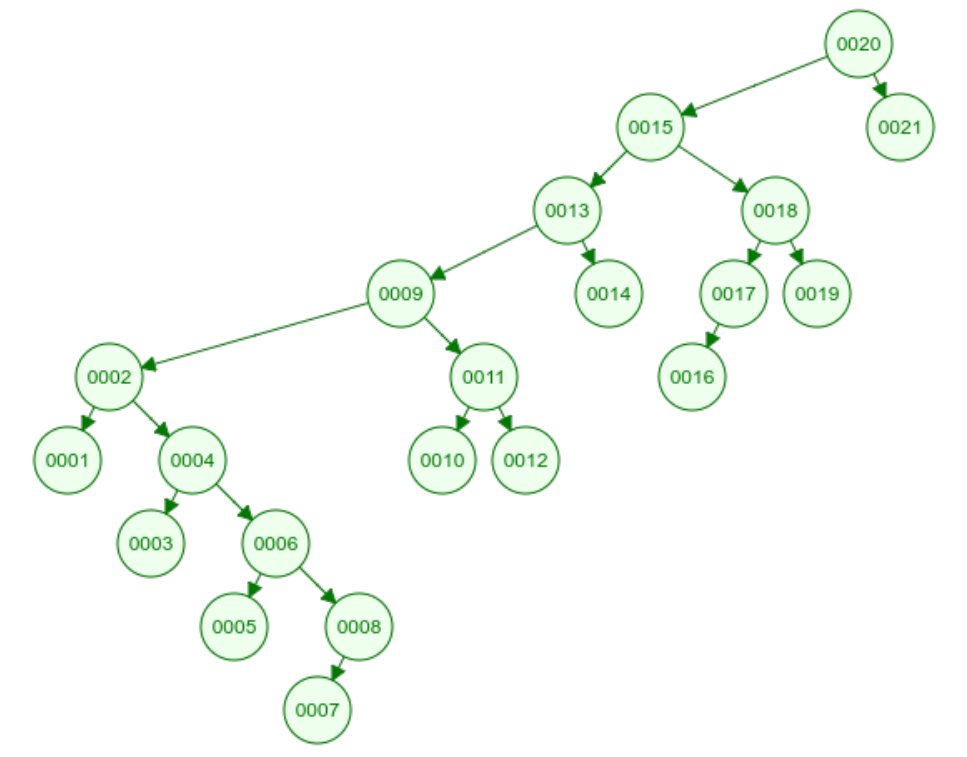
函数定义式：参量为 BinNode *x*,*y*,*T*，返回值为 BinNode 类型，函数名 **fingerSearch**

(a) 说明算法思路

(b) 写出伪代码

(c) 在图中画出由值为 6 的点查找值为 17 的点的查找路径

(d) 说明算法时间复杂度为 *O*(*d*)（若无法达到，说明困难在哪）



以下代码中的int parent[0,n)，是采用父结点表示法存储的任意一颗有根（但未必有序）的多叉树。（2014期中）

int f(int parent[],int n){

int h =-1;

for(int i=0;i<n;i++)

h=\_max(h,g(parent,i));

return h;

}

int g(int parent[],int i){

if(-1 == i) return -1;

return 1+g(parent,parent[i]);

}

A）以上算法f()和g()分别是何功能？

B）在最坏情况下，算法f()的渐进时间复杂度是多少？最坏情况何时出现？

C）在不做任何删除的前提下，试通过增加尽可能少的代码，使f()的运行时间降至O(n)，空间不超过O(n)。

简要说明你的改进策略与思路，然后直接在原代码基础上完成修改，并为关键环节增加注释。

（2020.1期末）

任给一棵二叉树S和二叉树T，设计一种时间复杂度为O(|S|+|T|)的算法，判断S是否与T的某棵子树同构（节点只存了rc, lc）。给出（1）算法的简要描述；（2）算法的具体步骤；（3）正确性的证明。

算法设计题（每小题 5 分，共 15 分）（2010年期末A卷-感觉此卷风格不是邓公）

设中序线索化二叉树的类声明如下：

template <class Type>

struct ThreadNode { //中序线索化二叉树的结点类

int leftThread, rightThread; //线索标志

ThreadNode<Type> \*leftChild, \*rightChild; //线索或子女指针

Type data; //结点中所包含的数据

};

template <class Type>

class inOrderThreadTree { //中序线索化二叉树类

public:

ThreadNode<Type> \* getRoot ( ) { return root; }

//其他公共成员函数

……

private:

ThreadNode<Type> \*root; //树的根指针

};

试依据上述类声明，分别编写下面的函数。

(1) ThreadNode<Type> \* getPreorderFirst (ThreadNode<Type> \*p);

//寻找以 p 为根指针的中序线索化二叉树在前序下的第一个结点。

(2) ThreadNode<Type> \* getPreorderNext (ThreadNode<Type> \*p)

//寻找结点\*p 的在中序线索化二叉树中前序下的后继结点。

(3) void preorder (inOrderThreadTree<Type>& T);

//应用以上两个操作，在中序线索化二叉树上做前序遍历。

**优先级队列-二叉堆、左式堆**

**判断**

**二叉堆**

在使用 Heapify 批量建堆的过程中，改变同层节点的下滤次序对算法的正确性和时间效率都无影响。（2016.1期末）

Floyd建堆，每次同层之间下滤顺序打乱，不影响复杂度和正确性。（2014期末补充）

相对于⼆叉堆，尽管多叉堆的⾼度更低，但⽆论是下滤⼀层还是整个下滤过程，时间成本反⽽都会增加。（2014期末）

完全二叉堆删除元素在最坏情况下时间复杂度为 *O*(log *n*)，但平均情况下仅为 *O*(1)。（2016.1期末）

多叉堆比二叉堆插入慢，删除快。（2014期末补充）

与二叉堆相比，多叉堆 delMax()操作时间复杂度更高。（2016.1期末）

相对于同样规模的完全二叉堆，多叉堆delMax() 操作的时间成本更低。（2019.1期末）

PFS每次调用priorUpdata()，总复杂度O(n) 。（2014期末补充）

PFS过程中，尽管每一步迭代都可能多次调用prioUdpater() ，但累计不过O(e)次。（2019.1期末）

若用完全二叉堆来实现PFS ，则各顶点在出堆之前，深度只可能逐步减少（或保持）而不致增加。（2019.1期末）

对于任何⼀颗⼆叉树T，其右、左⼦树的规模之⽐“λ=T.re().size()/T.le().size()”称作右偏率。对于（常规）⾼度同为h的AVL树（A），红⿊树（R），左式堆（L），若分别考察其λ所能达到的最⼤值，则在h⾜够⼤之后，三者按此指标的排列次序应是（ ）。（2014期末）

- A） L<R<A

- B） L<A<R

- C） R<A<L

- D） A<R<L

- E） 以上皆非

二叉堆中某个节点秩为 *k*，则其兄弟节点（假设存在）的秩为（ ）（2016.1期末）

A. *k* + 1

B. *k -*1

C. *k* + ( 1)^*k*

D. *k -* ( 1)^*k*

E. 以上皆非

 给定一个有 n 个数据元素的序列，各元素的值随机分布。若要将该序列的数据调整成为一个堆，那么需要执行的数据比较次数最多是多少？（2010年期末A卷-感觉此卷风格不是邓公）

**胜者树-败者树**

与胜者树相比，败者树在重赛过程中，需反复将节点与其兄弟进行比较。（2016.1期末）

胜者树的根节点即是冠军，而败者树的根节点即是亚军。（2019.1期末）

**左式堆**

对于左式堆 A 和 B，合并后所得二叉堆的右侧链元素一定来自 A 和 B 的右侧链。（2016.1期末）

采用Crane算法将左式堆A与B合并为左式堆H ，H右侧链上的节点未必都来自A或B的右侧链。（2019.1期末）

左式堆中每一对兄弟节点的高度尽管未必“左大右小”，但左兄弟至少不低于右兄弟的一半。（2019.1期末）

A B两个左式堆合并成H，H的右子树一定来自A或B的右子树？（2012.6期中）

有 2015 个节点的左式堆，左子堆最小规模为（ ）（不计外部节点）（2016.1期末）

A. 10

B. 11

C. 1007

D. 1008

E. 以上皆非

左式堆，左边一定大于等于右边的是（2012.6期中）

A.NPL B.规模 C.高度 D.外部节点数

**综合题**-左式堆（2014期末补充）

1）0,1,2.....2014，问左子树至少有几个节点，右子树最高多高（1,2012）

左子树至少1个节点，右子树最高2012

2）画出示意图

3）按什么顺序插入0..2014，能成为画的样子（2014,2013,0,1,2,...,2012）

**图**

**判断**

在图的优先级搜索过程中，每次可能调用多次 **prioUpdater**，但累计调用次数仍为*O*(*e*)。（2016.1期末）

如果把朋友圈视为一无向图，那么即使 A 君看不到你给 B 点的赞，你们仍可能属于同一个双联通分量。（2016.1期末）

设在有向图G中，存在⼀条自顶点v通往u的路径。于是，若在某次DFS中 有dTime(v) < dTime(u)，则这次DFS所⽣成的DFS森林中，v必定是u的祖先。（2014期末）

在图DFS() 算法中的default分支，将dTime(v)<dTime(u) 改为dTime(v)<fTime(u) 同样可行。（2019.1期末）

在⽆向连通图G中选定⼀个顶点s，并将各顶点v到s的距离记作dist(v)（特别地，dist(s)=0）。于是在G.Bfs(s)过程中，若辅助队列为Q，则dist(Q.front()) + 1 >= dist(Q.rear()) 始终成立。（2014期末）

我们知道，因同一顶点的邻居被枚举的次序不同，同⼀有向图G所对应的DFS森林未必唯⼀。然⽽只要起始于G中某顶点s的某次**DFS**所生成的是⼀棵树，则起始于s的任何⼀次**DFS**都将⽣成⼀棵树。（2014期末）

BFS、DFS的复杂度可能不是O(N+E)。（2014期末补充）

经过k条backward边就有k个环。（2014期末补充）

有向图经DFS后若共有k条边被标记为BACKWAR ，则它应恰有k个环路。（2019.1期末）

对于同一无向图，起始于顶点s的DFS尽管可能得到结构不同的DFS树，但s在树中的度数必然固定。（2019.1期末）

对于正权值有向图，如果把所有的边权都平方之后，Dijkstra算法得到的最短路径树方案不变。（2012.6期中）

**选择**

有向图的DFS不仅在起点任意，⽽且每⼀步迭代往往都会有多个顶点可供选择，故所生成的**DFS**森林并不唯⼀确定，且其中所含（ ）的数量也可能不同。（2014期末）

-A）树边

- B）前向边

- C）后向边

- D）跨越边

- E）以上皆⾮

**综合题**

在有向图 *G* 中，存在一条自顶点 *V* 通向 *u* 的路径，且在某次 DFS 中有 dTime[*v*]<dTime[*u*]，则在这次 DFS 所生成的 DFS 森林中，*v* 是否一定是 *u* 的祖先？若是，请给出证明；若不是，请举出反例。（2016.1期末）

有向图DFS遍历 (1x6+2x5=16)（2012.6期中）

给了一个7节点的有向图，节点标号为1~7，指定当存在歧义性的时候优先考虑标号小的节点。最后一共有6条树边(T)，1条跨边(C)，两条前向边(F)和两条后向边(B)，而且这题的分值写的是1\*6+2\*5=16，大家懂的。

**散列、hash**

**散列表**

若元素理想随机，则用除余法作为散列函数时，即使区间长度不是素数，也不会影响数据的均匀性。（2016.1期末）

采用单向平方策略的散列表，只要长度M是素数，则每一组同义词在表中都不会超过(M/2)(取下整)个。（2019.1期末）

将n个词条逐个插入一个容量为M 、采用线性试探策略、初始为空的散列表，n<M，则无论它们的插入次序如何，最终的平均成功查找长度都必然一样。（2019.1期末）

采用双向平方试探策略时，将散列表长度取作素数 *M* = 4*k* + 3，可以极大地降低查找链前 *M* 个位置冲突的概率，但仍不能杜绝。（2016.1期末）

我们知道，采取双向平⽅试探策略时，应该将散列表取作素数M = 4k + 3。尽管这样可以极⼤降低查找链前M个位置发⽣冲突的概率，但仍不能杜绝。（2014期末）

与 MAD 相比，除余法在（ ）有缺陷（2016.1期末）

A. 计算速度

B. 高阶均匀性

C. 不动点

D. 满射性

E. 以上皆非

相对于除余法，MAD法在（ ）方面有所改进。（2014期末）

- A）计算速度

- B）⾼阶均匀性

- C）不动点

- D）满射性

- E）以上皆⾮

（ ）属于针对闭散列策略的冲突排解方法。（2014期末）

- A）multiple slots

- B）linear probing

- C）overflflow area

- D）separate chaining

- E）quadratic probing

- F）double hashing

**综合题**

对闭散列 [0*, M*), *M* = 2*S*，采用如下冲突排列解决方法：（2016.1期末）

• 初始时，*c* = *d* = 0

• 探查 key 冲突时，*c ← c* + 1, *d ← d* + *c*，探查 H[(key+*d*)%*M*]

则这种算法是否可以保证空间能被 100% 利用？若是，请给出证明；若不是，请举出反例。

散列表（2014期末补充）

1、有若干数字，遵循模13、双向平方、懒惰删除，填写每一次操作后的散列表

2、最后问询问一个数字会发生什么情况（因为所有可用的都满了会产生死循环）

封闭散列（2019.1期末）

某散列表H[0,M=2^s) 采用封闭散列策略（初始令c=d=0 )：对于任何key ，首先试探H[key%M] ；以下，只要冲突，就令c <- c+1 再d <- d+c ，并继而试探H[(key+d)%M]  。以M=2^4=16 为例，关键码key=27 的

前五个试探位置依次是：11、12、14、1、5。但如同对于平方试探策略，我们首先需要确认，这种试探序列是否总能覆盖所有桶单元。若是，请给出证明；否则，试举一（s 和key组合的）反例。

散列表采用取余法+单向平方试探，M=7001，若某一时刻桶中仅有1481个数，却发生了重散列，为什么? 请简单说明，并大致绘制出表中当前状态。（2020.1期末-回忆版本1）

一个容量M=4079（素数）的哈希表，使用双向平方试探法，共插入了1231个词条，但在下一次插入时触发了rehash操作，问该哈希表此时的状态及rehash的原因。（2020.1期末-回忆版本2）

散列冲突 (20)（2012.6期中）

给定M=17的散列表，给定了基本策略：求余法、单向平方试探、懒惰删除。

进行了一系列操作，写出每次操作之后的散列表状态。

一开始put进去7个数，中间有一步put(1481)，第八个操作remove(1481)，最后一个操作put()一个数进去。

第一问，如果在上面操作之后查询1481，问将会出现什么情况。

第二问，在不改变基本策略的基础上，给出两种方案解决上述问题。

注意，不能改变那三个基本策略。

**跳转表**

跳转表期望高度O(logN)。（2014期末补充）

在 *n* 个节点的跳转表中，塔高的平均值为 *O*(log *n*)。（2016.1期末）

在存有n个词条的跳转表中，各塔高度的期望值为Θ(logn) 。

将硬币换成理想的骰子，且约定投出“6”时新塔才停止生长。于是对于同样存放n个元素的跳转表而言，（ ）的期望值将有所增长，但仍保持O(1）。（2014期末）

- A）查找过程中，在同⼀⾼度连续跳转的次数

- B）查找过程中，由“向右”到“向下”转折的次数

- C）查找过程中，沿同⼀座塔连续下⾏的层数

- D）（在查找定位之后）为创建⼀座新塔所需的时间

**串-KMP，BM-BC**

**判断**

若 KMP 算法不使用改进版的 next 表，最坏情况下时间复杂度可能达到 *O*(*mn*)。（2016.1期末）

在BM算法中，对于任⼀模式串P，0 < gs(j) <= j 对于每个0 <= j < |P| 都成⽴。（2014期末）

字符集变大，概率平均，则bc表比next表好。（2014期末补充）

相对于KMP算法而言，BM算法更适合于大字符集的应用场合。

无论是单独借助BC[]表或GS[]表，BM算法在最好情况下都只需要O(|T|/|P|)=O(n/m) 时间。（2019.1期末）

在KMP匹配的过程中，当主程序运行到i,j的状态时，意味着之前至少做过i次成功匹配以及i-j次失败匹配。（2012.6期中）

字符集规模越大的时候，next表比BC表效果越好。（2012.6期中）

**选择**

对小写字母集的串匹配，KMP 算法与蛮力算法在（ ）情况下渐进的时间复杂度相同（2016.1期末）

A. 最好

B. 最坏

C. 平均

D. 以上皆非

对随机生成的二进制串，gs 表中 gs[0]=1 的概率为（ ）（2016.1期末）

A. 1/2^*m*

B. 1/ 2^(*m-*1)

C. 1 /2^(*m*+1)

D. 1/*m*

**综合题**

写mamammi的改进后next表。（2014期末补充）

写ladygaga的GS表。（2014期末补充）

Next[]构造题 (8+6=14)（2012.6期中）

这题给了两个模式串，分别写出各自的改进前、后的next[]表

第一个串是SHIPSHIPED\_SHIPS，共16个字符

第二个串是PHILEDPHAL...，忘了最后是啥了，共12个字符

**排序**

**判断**

既然可以在 *O*(*n*) 时间内找出 *n* 个数的中位数，快速排序算法 (12-A1) 即可优化至 *O*(*n* log *n*)。（2016.1期末）

若序列中逆序对个数为 *O*(*n*2)，则使用快速排序 (12-A1) 须进行的交换次数为 *O*(*n* log *n*)。（2016.1期末）

⽆论g和h互素与否，已经**h-**有序的序列再经**g-**排序之后，必然继续保持**h-**有序。（2014期末）

shell排序若将插入排序改成归并排序，效率变快。（2014期末补充）

radix排序将桶排序改成quick排序，仍然正确 。（2014期末补充）

不存在CBA式算法，能够经过少于2n-3次比较操作，即从n个整数中找出最大和次大者。（2014期中）

存在CBA式算法，能够在O(n)时间内从n个无序整数中找出最大的10%。（2014期中）

起泡排序过程中，每经过一趟扫描交换，相邻的逆序对必然减少。（2014期中）

即便借助二分查找确定每个元素的插入位置，向量的插入排序在最坏情况下仍需Ω(n^2)时间。（2014期中）

经快速划分（LGU 版）之后，后缀G中的雷同元素可能调换相对次序，但其余部分的雷同元素绝不会。（2019.1期末）

只要底层的排序算法是正确且稳定的，则radixSort()也必然是正确且稳定的。（2019.1期末）

若输入序列包含Ω(n^2) 个逆序对，则快速排序算法（LUG 版）至少需要执行Ω(nlogn)  元素交换操作。（2019.1期末）

采用12-C节中介绍的任何一种增量序列，shellSort() 最后的1-sorting 都只需要O(n) 时间。（2019.1期末）

shellSort() 每按照某个增量做过逐列排序，序列中逆序对的总数都会减少（或持平），但绝不致增加。（2019.1期末）

一个向量的存在主流数，则该数必然是中位数以及频繁数。（似乎AB卷里的另一个是：如果有某数既是中位数又是频繁数，则该数也是主流数）（2012.6期中）

如果使用了线性复杂度的中位数选取算法，快速排序的复杂度可以保证在最坏情况下也渐进等于O(nlogn)。（2012.6期中）

希尔排序，如果序列已经g有序，换个参数再排一遍，依旧g有序。（2012.6期中）

若借助二分法查找确定每个元素的插入位置，向量的插入排序只需时间O(nlogn)时间。（2011期中）

只要是采用基于比较的排序算法，对任何输入序列都至少需要运行Ω(nlogn)时间。（2011期中）

**选择**

为从2014个随机元素中挑选出最⼤的5个，（ ）在最坏情况下所需的⽐较操作次数最少。（2014期末）

- A）构建⼤顶的锦标赛树，再做5次delMax()；

- B）⽤Floyd算法构建⼤顶堆，再做5次delMax()；

- C）采⽤选择排序算法，但仅执⾏前5次迭代；

- D）采⽤起泡排序算法，但仅执⾏前5次迭代；

- E）⽤linearSelect()算法找出第5⼤的元素，再遍历⼀趟找出（⾄多）4个⼤于它的元素

若仅考察最好情况下的渐进复杂度，则Bubblesort(p163版)，Insertionsort，Mergesort(p168+170版)，Selectionsort的非降排列次序是（）（2014期中）

A）IBMS   B）MIBS  C）SMIB   D）IMSB   E）BIMS

（）算法在最好情况下与最坏情况下的渐进性能相同。（2014期中）

A）Bubblesort(p163版)   B）Insertionsort   C）Mergesort(p168+170版)   D）Selectionsort

将有序列表L均分为长Θ(h)的k段，各段分别置乱，则 L.insertionSort()至多只需（）时间（2014期中）

A）Θ(h^2×k^2)   B）Θ(h×k^2)   C）Θ(h^2×k)  D）Θ(h×k)

**填空**：设在List::selectionSort()算法中，将：insertB(tail,remove(selectMax(head->succ,n))); （2014期中）

替换为：swap(tail->pred->data,selectMax(head->succ,n)->data);

若输入列表为{1962，1963，，，，2014；1，2，3，，，1960，1961}，

则swap()语句无实质效果(原地交换)的情况共计出现（ ）次。

**综合题**

K-选取（2014期末补充）

将1983个数字取前三大，最少比较多少次（锦标赛模型，答案貌似2013还2014）

shell排序（2014期末补充）

因为取1,2,4,8,...,2^n会产生最坏情况，因此每次取到偶数就+1或-1，问能不能避免最坏情况

1、每次遇偶数-1如何？

2、每次遇偶数+1如何？

长度为2020的序列仅能用交换进行排序，最坏情况下至少要交换几次? 为什么? （2020.1期末）

quickselect(A, k)中，给定n个数并随机置乱得到（无序）序列{a\_1, a\_2, … , a\_n}，

问（1）a\_i与a\_j进行比较的概率（用n, i, j, k表示）；（2）（a\_i与a\_j？或者n个数？）比较次数的期望；（3）？（2020.1期末）

就地堆排序 (15)（2012.6期中）

给了一个长度为7的随机整数向量，要求用Floyd算法建堆，然后排序。给了一个大表格，每一行都是向量的一个状态，第一行是建堆之后的状态，然后依次取最大值放到堆后面。

考察如下问题：任给 12 个互异的整数，且其中 10 个已组织为一个有序序列，现需要插入剩余的两个已完成整体排序。若采用基于比较的算法（CBA），最坏情况下至少需要做几次比较？为什么？ （2011期中）

向量的插入排序由 n 次迭代完成，逐次插入各元素。为插入第 k 个元素，最坏情况需要做 k 次移动，最好情况则无需移动。从期望的角度来看，无需移动操作的迭代次数平均有多少次？为什么？假定个元素是等概率独立均匀分布的。（2011期中）

**操作系统**

（20190520期末）

[ ] 实模式x86-32 CPU在冷启动后执行的第一条指令的地址由CS段基址加EIP确定，即FFFF\_FFF0H =FFFF\_0000H + 0000\_FFF0H。

（20190407期中）

[ ] 在x86-32中，在内核态通过对EFLAGS的设置可直接屏蔽时钟中断。

[ ] 在x86-32中，只有正确设定了段描述符表的相关配置，才能实现中断和异常处理。

[ ] 基于动态链接库的应⽤程序需要编译器能⽣成地址⽆关代码并需要OS加载库和程序。

[ ] 在ucore的bootloader代码实现中，在主引导记录 (master boot record) 格式下的512 字节的引导扇区 (boot sector) 中，除去最后两个字节 0x55AA, 还剩下 510 字节都能⽤来存放引导代码。

[ ] 在 lab1 中实现打印函数调⽤堆栈信息时， print\_debuginfo() 的参数不仅可以是 eip-1 ，还可以是 eip 或 eip-2 。##

**内存-连续内存、非连续内存**

（20190520期末）

[ ] 通过mmap系统调用，可以实现通过直接内存访问来访问文件，避免了read/write系统调用的特权级切换开销。

[ ] 访问频率置换算法(Frequency-based Replacement)综合采用了LRU和LFU算法用于磁盘缓存置换。

（20190407期中）

[ ] ⼆次机会（时钟）页面替换算法需要读取和修改页表项中访问位才能有效实现。

[ ] 运行在用户态的进程只能通过系统调用才能直接获得操作系统的服务。

[ ] 在内核态中执行的内核线程不需要拥有自己独占的页表。

[ ] 如没有页机制的虚存管理，则无法实现采用Copy on Write机制的fork系统调⽤。

[ ] 如果父进程比子进程先结束，那么子进程结束后将成为资源无法被回收的僵⼫进程。

1. 试说明描述全局和局部置换算法的不同，并分别各举出⼀种属于全局置换和局部置换的算法。（20190407期中）

2. 假定在⼀个虚拟存储系统中某进程分配了4个物理页面，当进程按**c, a, d, b, e, b, a, b, c, d**的序列进⾏⻚⾯访问时，当分别使用时钟置换算法与LRU算法时会出现多少次缺页？要求说明过程。（20190407期中）

3. 什么是Belady现象？请判断OPT、LRU、FIFO、Clock和LFU等各页面置换算法是否存在Belady现象？（20190407期中）

七、页面替换算法 （共 4 题 共 16 分） （20201028期末）

12.（4 分）

关于局部页面替换算法：最优算法（OPT）/先进先出算法（FIFO）/最近最久未使用算法（LRU）/时钟（clock）页面置换算法。

请问，什么是 Belady 现象？在上述四种算法中，哪些算法可能存在 Belady 现象？（回答字数 100 字）

13.（4 分）

关于局部页面替换算法：最优算法（OPT）/先进先出算法（FIFO）/最近最久未使用算法（LRU）/时钟（clock）页面置换算法。

请问，设一个页面访问序列为 0,1,4,2,3,4,1,0,3,2，且物理页帧中最多能容纳 4 个页面，初始时物理页帧为空，请用 FIFO 算法进行模拟，给出每一次访问的情况以及访问后物理页帧 的状态，最终给出总缺页次数。

14.（4 分）

关于局部页面替换算法：最优算法（OPT）/先进先出算法（FIFO）/最近最久未使用算法（LRU）/时钟（clock）页面置换算法。

请问，设一个页面访问序列为 0,1,4,2,3,4,1,0,3,2，且物理页帧中最多能容纳 4 个页面，初始时物理页帧为空，请用 LRU 算法进行模拟，给出每一次访问的情况以及访问后物理页帧 的状态，最终给出总缺页次数。

15.（4 分）

关于局部页面替换算法：最优算法（OPT）/先进先出算法（FIFO）/最近最久未使用算法（LRU）/时钟（clock）页面置换算法。

请问，从应用程序和操作系统在通用 CPU 上实际运行的角度看，如果操作系统采用 LRU 页面置换算法，那相比于采用 FIFO 页面置换算法，其在整体系统执行效率上的结果上一定更好吗？说明理由。（回答字数 150 字）

十、内存映射 （共 5 题 共 14 分） （20201028期末）

20.（6 分）

假定在某基于二级页表的虚拟页式存储，具有 4GB 物理内存的 32 位计算机系统下，32 位虚拟地址从高到低被划分如下， 10 位页目录序号、10 位页表序号和 12 位页内偏移。页目录和页表大小均为 4096 字节，页目录项和页表项大小为 4 字节。使用自映射方式组织页表。

本题表述中，用 `flags` 表示页表/页目录项的各种标志位，占低 12 位，且 CPU 不基于`flags`的信息来判断某项是页表项还是页目录项。要求回答中所有地址均使用用十六进制表示，如 `0x1edc8000`。 除了自映射，举出一个其他页表组织方式。与自映射相对比，你给出的其他页表组织方式有什么优缺点？要求至少一个优点和一个不足。（回答字数 150 字）

21-22.（共 4 分）

假定在某基于二级页表的虚拟页式存储，具有 4GB 物理内存的 32 位计算机系统下，32 位虚拟地址从高到低被划分如下， 10 位页目录序号、10 位页表序号和 12 位页内偏移。页目录和页表大小均为 4096 字节，页目录项和页表项大小为 4 字节。使用自映射方式组织页表。

本题表述中，用 `flags` 表示页表/页目录项的各种标志位，占低 12 位，且 CPU 不基于`flags`的信息来判断某项是页表项还是页目录项。要求回答中所有地址均使用用十六进制表示，如 `0x1edc8000`。

假定页目录起始物理地址是 0x1edc8000，并且物理地址 0x1edc8ff0 (=0x1edc8000+1020\*4)

开始的页目录项是自映射项。则此处自映射项的内容是 `\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_ | flags`，该自映射项的虚拟地址是`\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_`？

23-24.（共 4 分）

假定在某基于二级页表的虚拟页式存储，具有 4GB 物理内存的 32 位计算机系统下，32 位虚拟地址从高到低被划分如下， 10 位页目录序号、10 位页表序号和 12 位页内偏移。页目录和页表大小均为 4096 字节，页目录项和页表项大小为 4 字节。使用自映射方式组织页表。本题表述中，用 `flags` 表示页表/页目录项的各种标志位，占低 12 位，且 CPU 不基于`flags`的信息来判断某项是页表项还是页目录项。要求回答中所有地址均使用用十六进制表示，如 `0x1edc8000`。

假定虚拟地址 0x8af4b000 映射到物理地址 0x01572000，则位于虚拟地址 `\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_` 的页表项内容应该是 `0x01572000 | flags`。位于 0xff1bc500 的页表项，它指向的页的虚拟地址起始于 `\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_`。

**处理机调度、进程、线程管理**

（20190520期末）

[ ] 操作系统通过vfork创建子进程时，需要拷贝父进程的所有页表内容作为子进程的页表内容。

[ ] 信号量与管程都是高层同步互斥机制，无法基于信号量机制来实现管程。

[ ] 基于mesa机制的管程在执行条件变量的signal操作过程中，不会睡眠。

[ ] 基于hoare机制的管程在执行条件变量的signal操作过程中，不会睡眠。

[ ] 基于spinlock的互斥机制实现中，可以确保处于忙等线程集合将按线程忙等的等待时间顺序依次进入临界区。

[ ] 发生进程复制时，新进程的内核堆栈可以先于进程地址空间复制操作之前进行创建。

（20180525期末）

[ ] 在多CPU场景下，多个线程通过自旋锁(spinlock)争抢进入临界区执行，第一个成功进入临界区的线程是第一个执行自旋锁争抢的线程。

[ ] 运行在内核态的内核线程共享操作系统内核态中的一个页表。

[ ] 操作系统创建用户进程时需要为此用户进程创建一个内核栈用于执行系统调用服务等。

[ ] 通用操作系统的调度算法的主要目标是低延迟，高吞吐量，公平，负载均衡。

[ ] 单处理器场景下，短剩余时间优先调度算法(SRT)可达到具有最小平均周转时间的效果。

[ ] 单处理器场景下，无法通过打开和关闭中断的机制来保证内核中临界区代码的互斥性。

[ ] 信号量可用于解决需要互斥和同步需求的问题。

[ ] 属于管程范围的函数/子程序相互之间具有互斥性。

[ ] 操作系统处于安全状态，一定没有死锁；操作系统处于不安全状态，可能出现死锁。

[ ] 80386取指地址是base+eip，base是隐藏寄存器，初始化为0xffff0000，eip初始化为0xfff0，故执行的第一条指令是0xfffffff0。

[ ] 在x86-32 CPU下，操作系统可以实现让用户态程序直接接收并处理硬件中断。

[ ] 在操作系统中一旦出现死锁, 所有进程都不能运行。

[ ] 在ucore for x86-32中，子进程通过sys\_exit()执行进程退出时，ucore kernel会先释放子进程自身内核堆栈和进程控制块等，再唤醒父进程（或initproc），最后执行iret返回。

**12.** 某系统中共有M种类型资源，每种资源的个数为 R1，R2 ，R3 ，…，RM ，共有N个进程竞争使用这些资源，每个进程对这些资源的需求量为 K1，K2 ，K3 ，…，KM ，每个进程会依次获取这些资源，如下面的伪代码所示：（20190520期末）



当M=1，N=2时，在下面几种设定中，按照上面的获取方式，这些进程会发生死锁吗？

R1=11， K1=7 （---11---）

R1=11， K1=6 （---12---）

当M=1时，R1 ，N， K1 需要满足（---13--- ）不等式关系，才能使得这些进程不发生死锁。

（20180525期末）

（12分）设lab6中使用Stride调度算法，取BIGSTRIDE=100，假定各个进程的stride初始化为0。（注：lab6中的stride（32位整数，当前总共走了多少）和pass（32位整数，每一次走多少 pass=BIGSTRIDE/priority, 100>priority>1）的含义和论文原文含义相反）请回答下列问题。

A) 如果不考虑进程stride的值的溢出，那么对于任意两个进程A、B的stride值SA和SB，应当恒有abs(SA-SB) ≤ (\_\_1\_\_) ，为什么？

B) 考虑到abs(SA-SB)这一性质，假设stride值存在溢出，可将stride值的更新变为：stride = (stride + BIGSTRIDE/priority) mod n

那么，只要n> (\_\_2\_\_) ，那么stride算法就可以正常运行，为什么？

二、临界区 （共 2 题 共 4 分） （20201028期末）

2.（2 分）为了形成临界区，在单核机器上可以使用关中断，在多核机器上可以使用自旋锁。请回答下列问题。

在单核机器中使用自旋锁可以吗？为什么？（回答字数 150 字）

3.（2 分）为了形成临界区，在单核机器上可以使用关中断，在多核机器上可以使用自旋锁。请回答下列问题。

在多核机器上使用关中断可以吗？为什么？（回答字数 150 字）

三、进程管理 （共 4 题 共 10 分） （20201028期末）

4.（3 分）什么是孤儿进程？（回答字数 100 字）

5.（3 分）什么是僵尸进程？（回答字数 100 字）

6.（2 分）请描述在何种情况下有进程会成为孤儿进程。（回答字数 100 字）

7.（2 分）请描述在何种情况下有进程会成为僵尸进程。（回答字数 100 字）

四、多处理机调度 （共 2 题 共 6 分） （20201028期末）

8.（3 分）

在多处理机调度中，就绪任务队列的维护有两种策略：

单队列：使用无锁数据结构，维护一个全局队列

多队列：每个核有一个自己的局部队列

这两种方案相比，分别有什么好处和不足？（回答字数 150 字）

9.（3 分）在多处理机调度中，就绪任务队列的维护有两种策略：

单队列：使用无锁数据结构，维护一个全局队列

多队列：每个核有一个自己的局部队列

提出一种新的改进方案，避免这两种方案各自的不足之处。（回答字数 150 字）

五、嵌套异常 （共 1 题 共 8 分） （20201028期末）

10.（8 分） 由于内核实现代码的错误，导致内核在执行缺页（page fault）处理例程（响应用户进程导致的第一次缺页异常）的过程中再次（第二次）发生缺页异常。假定该内核其他部分编程正确且支持内核态中断。请描述第二次缺页异常产生之后的内核执行过程，以及从第一次缺页异常产生后开始的整个过程中内核栈变化和寄存器的状态变化的情况。（回答字数 200 字）

六、中断+调度+进程管理 （共 1 题 共 6 分） （20201028期末）

11.（6 分）假定在一台装有类 UNIX 通用操作系统的单核计算机上，任意一个中断从产生到响应处理结束，耗时 1 ms；OS 调度算法是时间片轮转调度算法，调度时间片为 100 ms ；时钟中断周期 10 ms；系统中有 3 个处于就绪态的进程。一个程序从十点整开始运行，恰好60,000 ms 后运行结束，在运行过程中产生了 6,100 次中断。 请根据以上信息回答：该程序的实际占用 CPU 的真实运行时间是否可以计算得出？若可以，给出计算过程和具体结果（单位 ms ，保留整数）；若不可以，给出理由。（回答字数 150字）

九、基于信号量的同步互斥 （共 1 题 共 14 分） （20201028期末）

19.（14 分）假设某仓库可存放 A、B 两种物品，其容量无限大；任何时刻只允许一个物品进行仓库进行入库操作；要求仓库中 A 物品数量 X 和 B 物品数量 Y 满足不等式： -M ≤ ( X - Y ) ≤ N ，其中 M 和 N 为正整数。请用信号量机制实现 A、B 两种物品入库的处理过程，要求能做到正确且高效的同步与互斥。

请说明所定义的信号量的含义和初始值，描述需要进行互斥处理的各种行为，描述需要进行同步处理的各种行为；要求用类 C 语言伪代码实现，并给出必要的简明代码注释。

**文件系统**

**11.** 请在下面有数字标号的空格中简要写出文件系统实现中文件分配的三种方式及其主要特点（需指明是否有外碎片问题、随机或顺序读取性能的好坏，可靠性差/好），并解释理由。（20190520期末）

连续分配的特点：文件顺序读取性能（---1---），随机读取性能（---2---），（---3---）外碎片。链式分配的特点：（---4---）外碎片，随机读取性能（---5---），可靠性（---6---）。索引分配的特点：（---7---）外碎片，随机访问性能（---8---），可靠性（---9---）。请用小于90个字给出对上面三类分配方式的填写内容的解释理由（---10---）。

**13.** 设初始情况下，一个一般文件fifile1的当前引用计数值为1，如果先建立fifile1的符号链接（软链接）文件fifile2，再建立fifile1的硬链接文件fifile3，然后再建立fifile3的硬链接文件fifile4。此时，fifile3和fifile4的引用计数值分别是 （---14---）和（---15--- ）。（20190520期末）

[ ] 由于符号链接（软链接）实际上是一类特殊的文件，它的内容就是其所指向的文件或目录的路径，所以符号链接可以指向一个不存在的文件或目录。（20180525期末）

[ ] 文件系统中，用于存储“文件访问控制信息”的合理位置是文件分配表。（20180525期末）

**磁盘、I/O**

[ ] 当前ucore lab8中实现的磁盘读写操作采用CPU轮询机制来实现。（20190520期末）

（20180525期末）

现有一个RAID磁盘阵列，包含6个磁盘，每个磁盘大小都是2TB，最大写入速度 200 MB/s，

最大读取速度 250 MB/s 的硬盘。用它们分别组成RAID级分别为0、1和5。假设在理想情况下（无中断、异常、预先缓存等外在干扰因素等），请回答下列问题。

A) 用它们组成的RAID0阵列的总可用空间为 (\_\_24.1\_\_)，最大写入速度为 (\_\_24.2\_\_)，最大读取速度为 (\_\_24.3\_\_)；

B) 用它们组成的RAID1阵列的总可用空间为 (\_\_24.4\_\_)，最大写入速度为 (\_\_24.5\_\_)，最大读取速度为 (\_\_24.6\_\_)；

C) 用它们组成的RAID5阵列的总可用空间为 (\_\_24.7\_\_)，最大写入速度为 (\_\_24.8\_\_)，最大读取速度为 (\_\_24.9\_\_)。

**计算机组成原理**

**指令、数据**

**判断**

（2016.1复习题）

加减交替法可以实现一位原码除法，比恢复余数法硬件实现简单。

奇偶校检可以发现并纠正一位错误。

（2010期末）

CISC 处理器的指令类型一般多于 RISC 处理器。

摩尔定律是指集成电路的频率每 18 个月翻一番。

微程序是供组合逻辑控制器来执行的一种机器语言程序。

能完成乘法运算的处理器都设置有乘法器。

（2019.1期末）

使用下面的程序初始化，判断前4题的对错。

in tx=random() ；

in ty=random() ；

in tz=random() ；

doubled x=(double) x；

doubled y=(double) y；

doubled z=(double) z；

1. ((x >> 1) << 1)<=x。

2. dx+dy+dz=(double) (z+y+x) 。

3. dx+dy+dz==dz+dy+dx。

4. dx\*dy\*dz==dz\*dy\*dx。

在32位的原码、反码和补码三种机器码中补码的表示范围最大。

CPI(Cycles Per Instruction) 用来衡量指令执行平均所需要的周期数， CPI越小， 代表执 行相同的程序所需要的时间越短。

只要运算器具有加法器和移位功能，再增加一些控制逻辑，就能实现乘除运算。

加减交替法可以实现原码除法，比恢复余数法硬件实现简单。

**填空**

（2016.1复习题）

Von Neumann 机中\_\_\_\_\_\_\_ 和 \_\_\_\_\_\_\_ 均以二进制形式存放在存储器中。

（2010期末）

给出十进制数-254 的 IEEE754 标准单精度浮点数表示。（-254）= \_\_\_\_\_\_\_ 。

某计算机字长 16 位，整数用补码表示，按字编址。某 C 语言定义了 i, j, k 共 3 个 short 型变量，其中有程序段如下：

{ i = 105; j = -12767; k = i + j; }。编译器将 i, j, k 三个变量分配到地址分别为 100、101 和 102 三个内存单元中。则上述程序段执行完成后，地址 100 内容为\_\_\_\_\_\_\_，地址 101 的内容为\_\_\_\_\_\_\_，地址 102 的内容为\_\_\_\_\_\_\_。（均用 16 进制表示）。

常见的指令寻址方式： \_\_\_\_\_\_\_、\_\_\_\_\_\_\_、\_\_\_\_\_\_\_ 、\_\_\_\_\_\_\_、 \_\_\_\_\_\_\_ 。

（2019.1期末）

十进制数66.375在IEEE 754单精度浮点数标准下表示为：\_\_\_\_\_\_\_。(16进制表示)。

使用海明码数据传输，设数据位k=3(D3D2D1)，校验位r=4(P4P3P2P1)，数据D3D2D1=011的编码结果为(按照P1P2D1P3D2D3P4的顺序) ：\_\_\_\_\_\_\_\_\_\_\_\_。

**选择**

（2016.1复习题）

以下哪个不是 Von Neumann 结构

A. ENIAC B. IBM 360 C. PDP-1 D. Pentium

布斯乘法中，是根据控制位和最低位的组合来判断功能的，若组合为 01 时，应该运算部分 积

A. +0 B. +[x]补 C. -[x]补 D. +[2x]补

ALU 是通过（ ） 逻辑电路实现的，其功能是（ ） 。

响应中断的流程包含 ：I. 存储 PC II. 保存所有通用寄存器III. 恢复 PC

A. 仅 I, III B.仅 I, II C. I, II, III D. 都不

相对于微程序控制器，硬布线控制器的特点是

A. 指令执行速度慢，指令功能的修改和扩展容易

B. 指令执行速度慢，指令功能的修改和扩展难

C. 指令执行速度快，指令功能的修改和扩展容易

D. 指令执行速度快，指令功能的修改和扩展难

某计算机的控制器采用微程序控制方式，微指令中的操作控制字段采用字段直接编码法， 共有 33 个微命令，构成 5 个互斥类，分别包含 7、3、12、5 和 6 个微命令，则操作控制字段至少 有

A. 5 位         B. 6 位         C. 15 位         D. 33 位

Von Neumann 机中指令和数据均以二进制形式存放在存储器中，CPU 区分它们的依据是

A. 指令操作码的译码结果

B. 指令和数据的寻址方式

C. 指令周期的不同阶段

D. 指令和数据所在的存储单元

下列关于 RISC 的说法错误的是

A. 寻址简单 B. 指令格式规范 C. 指令功能简单 D. 一般采用微程序实现

计算机的最小功能单元是

A. 字节 B. 程序 C. 微操作 D. 指令

（2010期末）

下列选项中，能缩短程序执行时间的是 。

I．提高 CPU 时钟频率 II．优化数据通路结构 III．对程序进行编译优化

A．仅 I 和 II     B．仅 I 和 III     C．仅 II 和 III     D．I，II，III

某一编码系统中，数据为 8 位。为提高系统的可靠性，希望能发现 2 位出错，并能在仅有 1 位出错时进行纠正，则需要增加的校验位的位数至少是 。

A．3 位 B．4 位 C．5 位 D．6 位

下列寄存器中，汇编程序员可见的是 。

A．存储器地址寄存器（MAR）

B．程序计数器（PC）

C．存储器数据寄存器（MDR）

D．指令寄存器（IR）

微程序存放的位置是 。

A．CPU     B．高速缓冲存储器     C．主存储器     D．磁盘存储器

（2019.1期末）

假定有4个整数用8位补码分别表示 r1=90H， r2=F2H， r3=FEH， r4=F8H， 若将运算结果存 放在一个8位的寄存器中，则下列运算会发生溢出的是

A.rl\*r2 B.r2\*r3  C.r3\*r4 D.r2\*r4

**简答**

（2016.1复习题）

指令和数据均存放在内存中，计算机如何从时间和空间上区分它们是指令还是数据？

（2019.1期末）

推导布斯乘法的规则，并依据规则使用5位的整数来计算3× (-6)。

在冯诺依曼结构计算机中，程序和数据都保存在内存中，那么处理器在 执行程序的过程中是如何区分程序和数据的?

**控制器-单多周期和流水线**

**判断**

（2010期末）

控制相关是指流水线的分支指令或其他需要改写 PC 的指令造成的冲突。

指令流水中使用旁路（Forwarding）技术可减少结构冲突。

**填空**

（2016.1复习题）

流水线中的相关可以分为 \_\_\_\_\_\_\_ 相关、 \_\_\_\_\_\_\_ 相关、\_\_\_\_\_\_\_ 相关。

五段流水线的指令执行步骤分别是取指令(IF)、\_\_\_\_\_\_\_ 、\_\_\_\_\_\_\_ 、\_\_\_\_\_\_\_ 、\_\_\_\_\_\_\_ 。

程序局部性原理包括 \_\_\_\_\_\_\_ 局部性和 \_\_\_\_\_\_\_ 局部性。

在五级流水线中，有以下三条 MIPS-16E 指令：ADDU R1 R2 R1，ADDU R1 R3 R2， ADDIU R3 2。

若不加入转发电路，则需要插入\_\_\_\_\_\_\_ 个气泡，如果加入转发电路需要\_\_\_\_\_\_\_ 个 气泡。本题中的三条指令使用了\_\_\_\_\_\_\_ 和的\_\_\_\_\_\_\_ 寻址方式。

（2019.1期末）

考虑指令序列：

SWR1，7(R2)             (指令1，R1寄存器的内容存到R2+7的地址)

ADDR5、R6、R3 (指令2，R5=R6+R3，寄存器值相加)

在一个5阶段流水线的MIPS处理器上， 第1个时钟周期执行指令1的取指阶段。指令1会发生访存地址未对齐异常，指令2会发生加法结果溢出异常、那么在第\_( )个时钟周期检测到访存地址未对齐异常，在第( )个时钟周期检测到加法结果溢出异常，最终处理器向操作系统报告( )异常。

使用我们实验用的板子设计MIPS处理器， 采用标准5级流水线无Cache设计， 每个时钟周期完成一个指令阶段。采用一片SRAM来保存数据与程序(另一片SRAM不使用) ，假设一个时钟周期可以完成SRAM的一次读或者一次写， 执行下面的指令序列：

LWR1，0(R2)(指令1，从R2+0位置读入数据到R1中)

SUBR4，R1，R5(指令2，R4=R1-R5)

ANDR6，R2、R7(指令3，R6=R2&R7)

ADDR8，R3，R2(指令4，R8=R3+R2)

ADDR9，R1，R2(指令5，R9=R1&R2)

SWR9，I6(R2)(指令6，将R9的数据存入到16+R2的位置)

若处理器设计时没有数据旁路，执行完所有的指令需要\_( )个时钟周期，增加数据旁路后，执行完所有的指令需要( )个时钟周期。

**选择**

（2016.1复习题）

不可用于解决控制冲突的是

A. 插入等待 B. 延迟槽 C. 数据转发 D. 分支预测

一台有完整的层次储存器的 MIPS 计算机，LW 指令访存的最少次数为

A. 0 B.1 C. 2 D. 3

（2010期末）

下面关于多周期 CPU 的描述，正确的是 。

A．指令周期长度固定

B．每个机器周期可完成一条指令

C．多个机器周期完成一条指令

D．其控制器只用组合逻辑电路就能实现

（2019.1期末）

一台有完整的层次储存器的MIPS计算机， 执行LW指令访问主存的最少次数为

A.0  B.1  C.2  D.3

下列关于通过数据转发来解决数据冲突的描述中、正确的是

(流水线阶段标识：EX为执行阶段、MEM为访存阶段， WB为写回阶段)

A.只能转发EX MEM流水线寄存器到ALU的输入

B.只能转发MEM/WB流水线寄存器到ALU的输入

C.在EX/MEM流水线寄存器和MEM/WB流水线寄存器对应的寄存器都进行转发的时候、选择MEM WB流水线寄存器的值

D.在EX MEM流水线寄存器和MEM WB流水线寄存器对应的寄存器都进行转发的时候、选择EX/MEM流水线寄存器的值

**简答**

（2016.1复习题）

数据旁路的含义和目的？

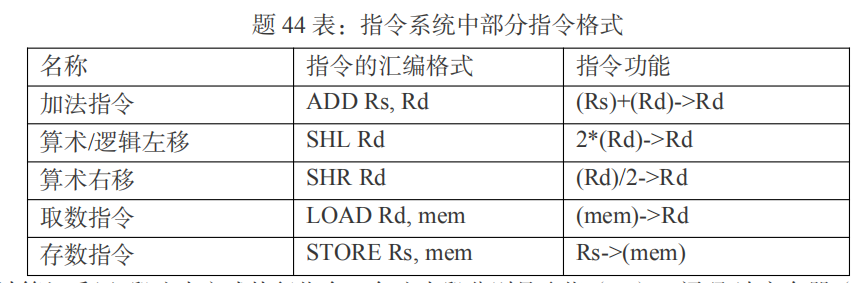
什么是流水线中的结构冲突？MIPS 中在哪些流水阶段中会发生结构冲突？对应的解决途径都有哪些？

请说明指令周期、机器周期、时钟周期之间的关系。

**综合分析题**

（2016.1复习题）

某 16 位计算机中，带符号整数用补码表示，数据 Cache 和指令 Cache 分离。题 44 表给出了指令系统中部分指令格式，其中 Rs 和 Rd 表示寄存器，mem 表示存储单元地址，（x）表示寄存器 x 或存储单元 x 的内容。



该计算机采用5段流水方式执行指令，各流水段分别是取指（IF）、译码/读寄存器（ID）、执行/计算有效地址（EX）、访问存储器（M）和结果写回寄存器（WB），流水线采用“按序发射， 按序完成”方式，没有采用转发技术处理数据相关，并且同一寄存器的读和写操作不能在同一个时钟周期内进行。请回答下列问题。

（1）若 int 型变量 x 的值为-513，存放在寄存器 R1 中，则执行“SHL R1”后，R1 中的内容是多少？（用十六进制表示）

（2）若在某个时间段中，有连续的 4 条指令进入流水线，在其执行过程中没有发生任何阻塞， 则执行这 4 条指令所需的时钟周期数为多少？

（3）若高级语言程序中某赋值语句为 x=a+b，x、a 和 b 均为 int 型变量，它们的存储单元地址分别表示为[x]、[a]和[b]。该语句对应的指令序列及其在指令流中的执行过程如题 44 图所示。



请解释为什么I3 的 ID 段被阻塞？为什么I4 的 IF 段被阻塞？

（4）若高级语言程序中某赋值语句为 x=x\*2+a，x 和 a 均为 unsigned int 类型变量，它们的存储单元地址分别表示为[x]、[a]，则执行这条语句至少需要多少个时钟周期？要求模仿题 44 图画出这条语句对应的指令序列及其在流水线中的执行过程示意图。

**存储器**

**判断**

（2016.1复习题）

直接映射的 Cache 可以使用 LRU 或 FIFO 的替换方式。

动态预测可以增加 Cache 命中率。

段式虚存中的段表存有段长信息，可以检查是否访问地址越界。

分页系统中增加 TLB 可以提高命中率。

容量为 128 字节、采用直接映射方式 Cache 的缺失率和容量为 64 字节、采用 2 路组相联映射方式 Cache 的缺失率相当。

硬盘不同的磁记录方式对于存储容量没有影响。

FLASH 和 SRAM 一样都是电易失性存储器。

DMA 可以提高硬盘到内存的载入速率。

（2010期末）

RAID5 和 RAID4 比较，检错纠错能力更高。

Cache 总容量一定的话，两路组相连组织方式的命中率不低于直接映射方式的命中率。

页式虚拟存储器管理中设置 TLB 可提高访问速度。

虚拟存储管理中，虚页数和实页数必须相同。

（2019.1期末）

处理器缓存Cache与存储器的地址的映射关系是由操作系统来管理的。

使用高速缓存是为了提高主存的容量。

**填空**

（2016.1复习题）

页式存储，在\_\_\_\_\_\_\_ 中设置\_\_\_\_\_\_\_ 进行虚实转换。

Cache 的缺失种类有：\_\_\_\_\_\_\_ 、\_\_\_\_\_\_\_ 、\_\_\_\_\_\_\_ 、无效缺失。

（2019.1期末）

为了存储256GB的数据， 使用RAID0的方式， 所占用的磁盘空间大小为\_\_\_\_\_\_\_， 使用RAID 6的方式(4+2) ， 所占用的磁盘空间大小为\_\_\_\_\_\_\_。

**选择**

（2016.1复习题）

假定不采用 Cache 和指令预取技术，且机器处于开中断状态，则在下列有关指令执行的叙述中，错误的是

A. 每个指令周期中 CPU 都至少访问内存一次

B. 每个指令周期一定大于或等于一个 CPU 时钟周期

C. 空操作指令的指令周期中任何寄存器的内容都不会被改变

D. 当前程序在每条指令执行结束时都可能被外部中断打断

直接映射 Cache 中，命中率最高的算法

A. FIFO B. LRU C. RAND D. 都不对

设计一个字长 16 位，容量为 32KW 的内存，需要用几片 2K × 8 bit 的存储芯片

A. 16 B. 32 C. 64 D. 128

下列关于 Cache 与 TLB 的描述中，哪个说法是错误的

A. TLB 与 Cache 中保存的数据是不同的

B. TLB 缺失之后，有可能直接在 Cache 中找到页表内容

C. TLB 缺失会导致程序执行出错，但是 Cache 缺失不会

D. TLB 和 Cache 的命中率都与程序的访存模式有关

下列有关 RAM 和 ROM 的叙述中，正确的是

I. RAM 是易失性存储器，ROM 是非易失性存储器

II. RAM 和 ROM 都采用随机存取方式进行信息访问

III. RAM 和 ROM 都可用作 Cache

IV. RAM 和 ROM 都需要进行刷新

A. 仅 I 和 II B. 仅 II 和 III C. 仅 I、II 和 IV D. 仅 II、III 和 IV

假设某计算机按字编址，Cache 有 4 个行，Cache 和主存之间交换的块为 1 个字。若 Cache 的内容初始为空, 采用 2 路组相联映射方式和 LRU 替换算法。当访问的主存地址依次为 0,4,8,2,0,6,8,6,4,8 时，命中 Cache 的次数是

A. 1         B. 2         C. 3         D. 4

（2010期末）

下面关于高速缓冲存储器（Cache）的᧿述中，错误的是 。

A．高速缓冲存储器设置在主存和 CPU 之间

B．高速缓冲存储器由系统程序员编程管理

C．高速缓冲存储器访问速度高于主存储器

D．高速缓冲存储器以块为单位和主存交换数据

下面有关静态存储器（SRAM）和动态存储器（DRAM）的叙述中，正确的是 。

I． SRAM 和 DRAM 都是电易失性存储器

II． SRAM 和 DRAM 都采用随机存取方式进行数据访问

III． SRAM 和 DRAM 都可用作 Cache

IV． SRAM 和 DRAM 都需要刷新

A．仅 I 和 II     B．仅 I 和 III     C．仅 II 和 IV     D．仅 III 和 IV

下面的命中和缺失的组合情况，在一次访存过程中，不可能发生的是 。

A．TLB 未命中，页表未命中，Cache 未命中

B．TLB 未命中，页表命中，Cache 命中

C．TLB 命中，页表命中，Cache 未命中

D．TLB 命中，页表未命中，Cache 命中

某磁盘有 100 个柱面，每个柱面有 10 个磁道，每个磁道有 128 个扇区，每个扇区容量为 512 字节。该磁盘的存储容量是 。

A．12800B     B．25MB     C．62.5MB     D．625MB

（2019.1期末）

下面不属于缓存缺失的情况有 \_\_\_\_\_

A.必然缺失 B.容量缺失 C.页面缺失 D.冲突缺失

某计算机主存容量为128KB， 其中ROM区为8KB， 其余为RAM区， 按字节编址。现要用2K×8位的ROM芯片和4K×4位的RAM芯片来设计该存储器， 则需要上述规格的ROM芯片数和RAM芯片数分别是

A.2、30      B.4、30      C.2、60      Ｄ.4、60

假定不采用Cache和指令预取技术， 且机器处于开中断状态， 则在下列有关指令执行的叙述中，错误的是

A.每个指令周期中CPU都至少访问内存一次

B.每个指令周期一定大于或等于一个CPU时钟周期

C.空操作指令的指令周期中任何寄存器的内容都不会被改变

D.当前程序在每条指令执行结束时都可能被外部中断打断

下列关于Cache与TLB的描述中， 说法错误的是

A.TLB与Cache中保存的数据是不同的

B.TLB缺失之后， 有可能直接在Cache中找到页表的内容

C.TLB缺失会导致程序执行出错， 但是Cache缺失不会

D.TLB和Cache的命中率都与程序的访存模式有关

**简答**

（2016.1复习题）

除了采用高速芯片外，分别指出存储器、运算器、控制器和 I/O 系统各自可采用什么方法提高机器速度，各举一例简要说明。

在 DMA 方式中，CPU 和 DMA 接口分时使用主存有几种方法？简要说明其原理和特点。

（2010期末）

简述一次 DMA 传送的过程。

（2019.1期末）

某计算机系统的内存系统中， 已知Cache访问时间为20ns， 主存访问时间为100ns。CPU执行一段程序时， CPU访问内存系统共5000次， 其中访问主存的次数为450次。那么Cache命中率是\_\_\_\_\_\_\_ ，CPU访问内存的平均访问时间是\_\_\_\_\_\_\_ 。

假定一台计算机的显示存储器用DRAM芯片实现， 若要求显示分辨率为800\*600， 使用RGB颜色， 每种颜色使用1个字节表达， 帧频为60Hz， 显存总带宽的50%用来刷新屏幕，则需要的显存总带宽至少约为\_\_\_\_\_\_\_ 。

假设流水线寄存器或者锁存器的输出延迟为100ps，输入延迟可以不考虑，不使用处理器内部的缓存(Cache) 。各阶段的的延迟如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **IF** | **ID** | **EX** | **MEM** | **WB** |
| 250ps | 180ps | 150ps | 300ps | 200ps |

实现以下7条指令：

1. addu  rd rs rt，

2. subu rd rs rt，

3. ori rt rs imm，

4. lw  rt， rs， imm，

5. sw  rt， rs， imm，

6. beq  rs， rt， imm，

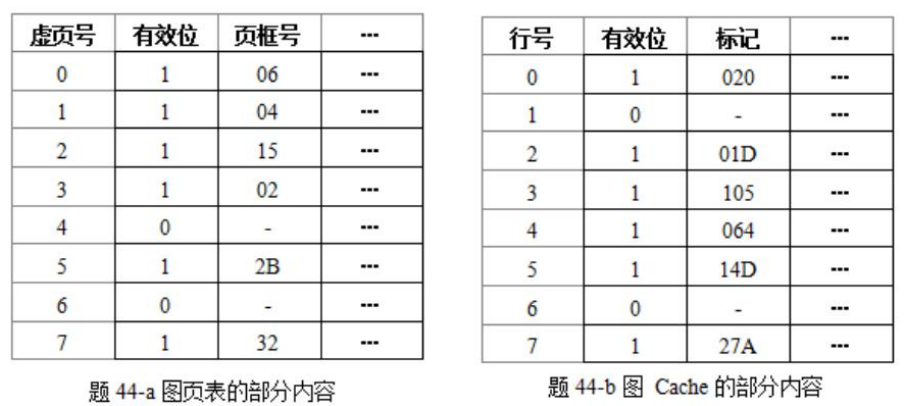
7. j target

如果设计为单周期的处理器，则处理器的时钟周期至少为\_( )的处理器，则处理器的时钟周期至少为\_\_\_\_\_\_\_ ，多周期处理器最长指令延迟为\_\_\_\_\_\_\_ ；如果设计为多周期( )如果设计为标准的五阶段流水线，处理器的时钟周期至少为\_\_\_\_\_\_\_ 。

**综合分析题**

（2016.1复习题）

某计算机存储器按字节编址，虚拟（逻辑）地址空间大小为16MB，主存（物理）地址空间大小为1MB，页面大小为4KB；Cache 采用直接映射方式，共8 行；主存与Cache 之间交换的块大小为32B。系统运行到某一时刻时，页表的部分内容和Cache 的部分内容分别如题44-a 图、题44-b 图所示，图中页框号及标记字段的内容为十六进制形式。请回答下列问题。



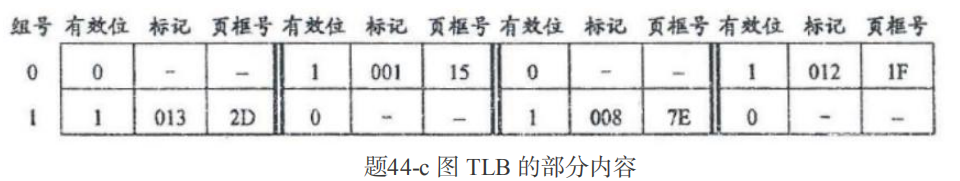
(1) 虚拟地址共有几位，哪几位表示虚页号？物理地址共有几位，哪几位表示页框号（物理页号）？

(2) 使用物理地址访问Cache 时，物理地址应划分成哪几个字段？要求说明每个字段的位数及在物理地址中的位置。

(3) 虚拟地址001C60H 所在的页面是否在主存中？若在主存中，则该虚拟地址对应的 物理地址是什么？访问该地址时是否Cache 命中？要求说明理由。

(4) 假定为该机配置一个4路组相联的TLB 共可存放8个页表项，若其当前内容（十六进制）

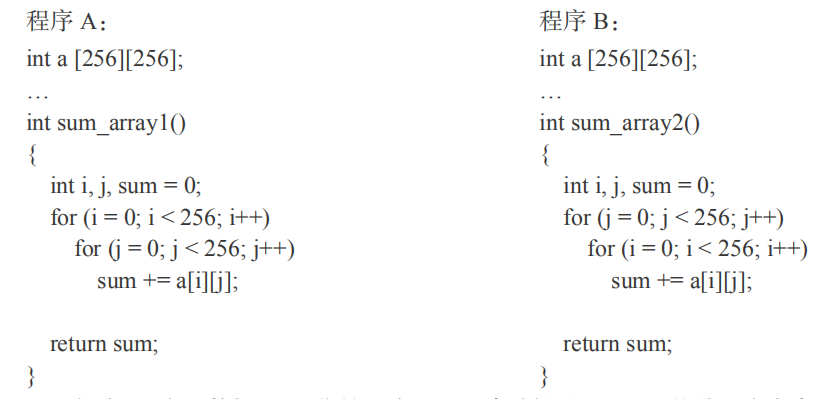
如题44-c图所示，则此时虚拟地址024BACH所在的页面是否存在主存中？要求说明理由。



3. 某计算机的主存地址空间大小为 256 MB，按字节编址。指令 Cache 和数据 Cache 分离，均

有 8 个 Cache 行，每个 Cache 行大小为 64 B，数据 Cache 采用直接映射方式。现有两个功能相同

的程序 A 和 B，其伪代码如下所示：



假定 int 类型数据用 32 位补码表示，程序编译时 i, j, sum 均分配在寄存器中，数组 a 按行优先方式存放，其首地址为 320（十进制数）。请回答下列问题，要求说明理由或给出计算过程。

（1）若不考虑用于 Cache 一致性维护和替换算法的控制位，则数据 Cache 的总容量为多少？

（2）数组元素 a[0][31]和 a[1][1]各自所在的主存块对应的 Cache 行号分别是多少（Cache 行号从 0 开始）？

（3）程序 A 和 B 的数据访问命中率各是多少？哪个程序的执行时间更短？

硬盘的寻道时间是 8ms，转速为 7200RPM，传输速率 5MB/s，每个磁道有 64 个盘区，每个盘区大小 512 字节，控制器延迟为 1.5ms。求：

(1) 读单盘区的时间；

(2) 读连续的 8KB 的时间；

(3) 假如我们有 4 个磁盘能并行的读出数据，那么读取 32KB 需要多少时间？

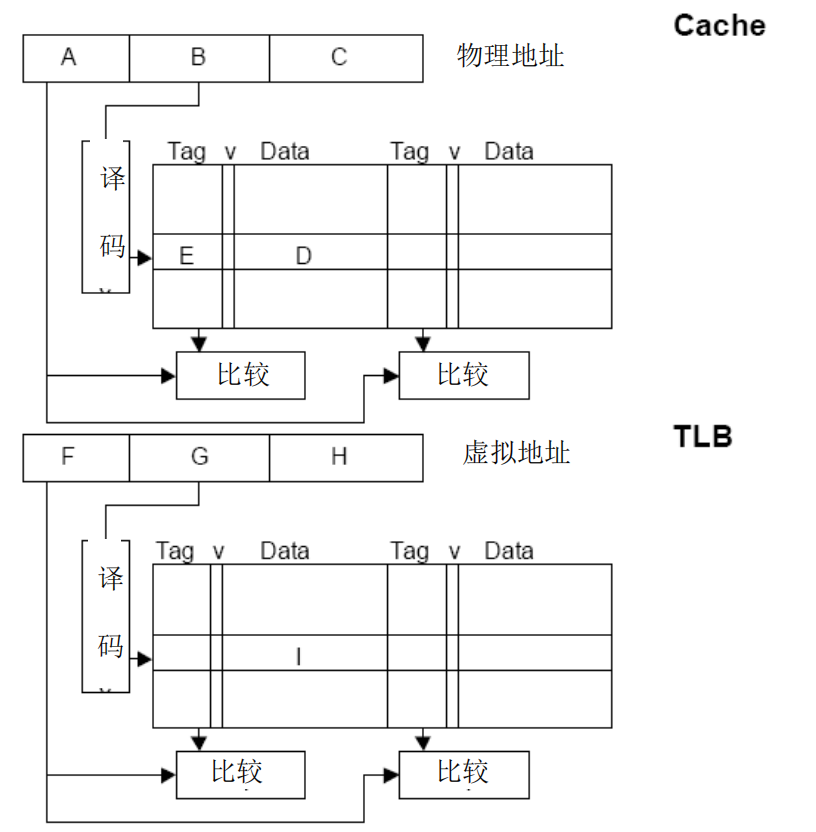
（2010期末）

某计算机存储器系统参数如下：

* TLB 共有 256 项，按两路组相连方式组织；
* 64KB 的数据 Cache，块大小为 64B，组织方式也是两路组相连；
* 虚拟地址 32 位，物理地址 24 位；
* 页大小为 4KB。

下图给出了系统的简单示意。

请分别计算其中各字段 A、B、C、D、E、F、G、H 和 I 所占的位数，给出计算过程。



**I/O、中断、总线**

**判断**

（2010期末）

PCI Express 总线和 PCI 总线结构基本相同，只是ᨀ高了总线频率，带宽因此得到了提高。

同步传输总线使用统一的时钟来协同总线事务。

Blu-Ray 采用的激光波长比 DVD 的波长更短。

**填空**

（2016.1复习题）

I/O 通道的类型分别是\_\_\_\_\_\_\_ 、\_\_\_\_\_\_\_ 、\_\_\_\_\_\_\_ 。

（2010期末）

中断处理过程包括关中断、\_\_\_\_\_\_\_ 、\_\_\_\_\_\_\_、开中断、\_\_\_\_\_\_\_、关中断、\_\_\_\_\_\_\_ 、开中断、\_\_\_\_\_\_\_等步骤。

（2019.1期末）

MIPS响应异常和中断的硬件流程包含 I.保存PC寄存器 Ⅱ.保存所有通用寄存器 III.恢复PC寄存器 IV.保存异常原因 V.读取异常原因

A.I，II，III，     B.II，III，IV， C.I，III，IV      D.III，IV，Ⅴ

**选择**

（2016.1复习题）

下列关于 USB 总线特性的描述中，错误的是

A. 可实现外设的即插即用和热拔插

B. 可通过级联方式连接多台外设

C. 是一种通信总线，连接不同外设

D. 有 2 根数据线，可同时传输 2 位数据，数据传输率高

五个中断，响应优先级为 0>1>2>3>4，处理优先级为 4>0>2>1>3，问 1 的中断屏蔽字（顺序为 43210）

A. 11110 B. 01101 C. 00011 D. 01010

下列关于中断 I/O 方式和 DMA 方式比较的叙述中，错误的是

A. 中断 I/O 方式请求的是 CPU 处理时间，DMA 方式请求的是总线使用权

B. 中断响应发生在一条指令执行结束后，DMA 响应发生在一个总线事务完成后

C. 中断 I/O 方式下数据传送通过软件完成，DMA 方式下数据传送由硬件完成

D. 中断 I/O 方式适用于所有外部设备，DMA 方式仅适用于快速外部设备

20. 下列选项中，在 I/O 总线的数据线上传输的信息包括 I. I/O 接口中的命令字 II. I/O 接口中的状态字 III.中断类型号

A. 仅 I、II B. 仅 I、III         C. 仅 II、I1II         D. I、II、III

（2010期末）

某总线采取菊链仲裁方式，则下列关于总线优先级的描述中，正确的是 。

A．连接在该总线上的设备得到总线授权的优先级相同。

B．越靠近总线仲裁器的设备优先级越高。

C．越远离总线仲裁器的设备优先级越高。

D．响应速度越快的设备优先级越高。

（2019.1期末）

假定某计算机的CPU主频为1GHz、外接外设接口中有一个16位的数据缓存器， 中断服务程序的执行时间为500个时钟周期，则中断能够处理的最大传输带宽为

A.3MB/sB.4MB/sC.5MB/sD.6MB/s

**简答**

（2016.1复习题）

什么是总线？总线仲裁是什么？总线仲裁的两种方式为？

**计算机网络**

**物理层**

**选择**

（2018期末）

物理层四个特性

（2012期末）

下列关于信道叙述错误的是：

a. 没记住    b. 没记住    c. 没记住

d. 信噪比 20dB 即为信号功率除以噪声功率等于 20。

定义物理层工作规程与时序的是物理层的哪个特性？

a. 电气特性    b. 机械特性    c. 规程特性    d. 功能特性

**数据链路层**

**选择**

（2018期末）

数据链路层的基本功能

1-坚持、非坚持、p 坚持的含义

PPP 协议的特点

10M 以太网的波特率

（2012期末）

下列关于交换技术叙述错误的是：

a. 电路交换在发送与接收方的物理链路上预留带宽

b. 虚电路交换的分组头部需要全局地址信息

c. 数据报交换可能出现分组乱序

d. 报文交换要求有较大缓存

停等协议效率最低的是？

后退 N 帧协议，发送了 0～7 号帧，发送方定时器超时时收到了 0，2，3 号 ACK，发送方需要重发几个帧？

a. 2      b. 3      c. 4      d. 5

下列有关 PPP 协议叙述错误的是：

a. 动态分配 IP 地址

b. 面向比特的协议

c. 使用 NCP 协商

d. 支持身份认证

先监听，若忙则等待随机一段时间在发送的是什么 MAC 层协议？

a. 四个选项都忘了……

数据链路层提供的基本控制功能是？

a. 差错控制    b. 顺序控制    c. 流量控制    d. 拥塞控制

使用位填充方法，以 01111110 为位首 flag，数据为 011011111111111111110010，求问传送时要添加几个 0？

a. 1      b. 2      c. 3      d. 4

10001001 与 10110101 的海明距离？

在 802.3 标准中，发送帧之前需要：

a. 等待冲突    b. 等待令牌    c. 监听介质    d. 接受一个帧

下列有关无线局域网的叙述，错误的是：

a. 实现了载波监听

b. 冲突被发送站发现

c. 使用 MACA 机制

d. 某时刻信道有多个有效数据帧

10Base-T 以太网的最大网段距离：

a. 2000m      b. 500m      c. 200m    d. 100m

能从 MAC 地址解析出 IP 地址的协议是？

a. ICMP      b. PPP      c. ARP      d. RARP

**简答**

（2013期末）

以太网的最大最小帧长，为什么要有？

802.3 和 802.11 协议的异同

Rdt3.0 提供了哪些可靠的服务？

循环冗余检测，给定  D=1010101010，G=10011， 求 R

**综合题**

（2018期末）

请比较一下在一个电路交换网络中和在一个（负载较轻的）分组交换网络中，沿着k跳的路径发送一个x位消息的延迟情况。电路建立的时间为s秒，每一跳的传播延迟为d秒，分组的大小为p位，数据传输率为b bps。在什么条件下分组网络的延迟比较短？

假定x位用户数据将以一系列分组的形式，在一个分组交换网络中沿着一条共有k跳的路径向前传输，每个分组包含p位数据和h位的头，这里x>>p+h。线路的传输率为b bps，传播延迟忽略不计。请问，什么样的p值使总延迟最小？

**网络层**

**选择**

（2018期末）

IP、ARP 协议的特点：

IPV4 和 IPV6 都只对头部校验？

ARP 解析的总是默认网关？

交换方式中哪些属于存储转发

（2012期末）

下列关于网络体系结构叙述错误的是：

a. 计算机网络体系结构是协议的集合

b. 在分层模型中，上层只知道下层的服务，不知道实现

c. 网络体系结构中最广泛使用的是 TCP/IP 模型

d. 同层对等实体的信息交换规则成为协议。

分层网络体系结构中，N 层收到 N+1 层 SDU 之后的操作是：

a. 加上 PCI，生成 PDU

b. 剥除 PCI，生成 PDU

c. 加上 ICI，生成 PDU

d. 剥除 ICI，生成 PDU

90 字节的 IP 分组封装到以太网中，需要填充多少个字节？

a. 38      b. 10      c. 6      d. 0

ICMP 属于哪层协议？

a. 数据链路层    b. 网络层    c. 传输层    d. 应用层

**简答**

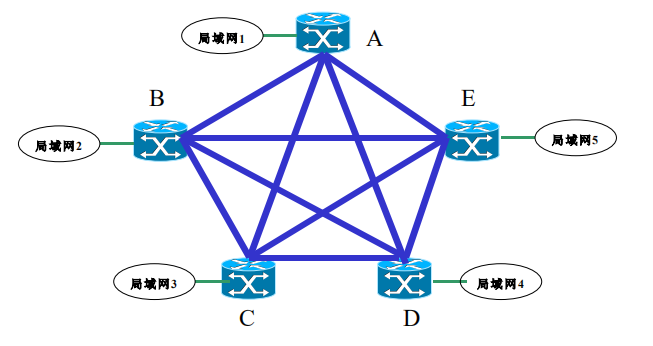
（2013期末）

IPv6 和 IPv4 的区别？前者的偏移地址是多少位？报头格式？

**综合题**

（2018期末）

如图所示，5台路由器组成全相连的网络，每台路由器有5个接口，分别连接其它4台路由器和1个局域网，每个局域网最多连接20台计算机，每台计算机分配1个IP地址。如果只有一个IPv4地址块202.112.10.0/24可供分配，请给出一种合理的地址分配方案，分别给出每个局域网的地址空间和路由器每个端口的地址以及它们的掩码。



假设主机A被连接到一台路由器R1上，R1又连接到另一台路由器R2上，R2被连接到主机B。假定一条TCP消息包含900字节的数据和20字节的TCP头，现在该消息被传递给主机A的IP地址，请它递交给主机B。请写出在三条链路上传输的每个分组中IP头部的Total length，Identification，DF，MF和Fragment offset域。假定链路A-R1可以支持的最大帧长度为1024字节，其中包括14字节帧头；链路R1-R2可以支持最大帧长度为512字节，其中包括8字节的帧头；链路R2-B可以支持的最大帧长度为512字节，其中包括12字节帧头。

**传输层**

**选择**

（2018期末）

面向连接和面向无连接服务的含义

服务原语的四种类型

滑动窗口协议中，若窗口序号由 4 位表示，发送窗口与接受窗口大小相同，则发送窗口最大多大

HTTP1.0 中，传输一个文本和三个图片需要建立多少 TCP 连接

一个 TCP 段在 IP 层传输时分成两个包，长度分别为 500，480，问 TCP 段原来长度

给出子网掩码，问能容纳多少主机

（2012期末）

TCP 端对端通信作用于：

a. 主机之间    b. 网络之间    c. 进程之间    d. 主机到网络

TCP 使用滑动窗口协议实现：

a. 端到端流量控制

b. 全网控制

c. 端到端流量和拥塞控制

d. 差错控制

对于传输层来说错误的是：

a. TCP 是全双工协议

b. TCP 是字节流协议

c. TCP 和 UDP 协议不能使用同一个端口

d. TSAD 是 IP 和端口的组合

对于 UDP 协议，如果想实现可靠传输，应在那一层实现？

a. 数据链路层    b. 网络层    c. 传输层    d. 应用层

TCP 使用慢启动算法，为了

a. 减小拥堵

b. 高速传输

c. 快速探测网络承载力

d. 适应接收窗口的大小

**简答**

（2013期末）

N=4，编号为 0，1，2，3，…，10，发送几个分组的时延为 2RTT，定时重传时延为 0.4RTT.

1. GBN：只有分组 2 在发送时丢失，求接下来 N 个发送分组的编号

2. SR：只有分组 2 在发送时丢失，求接下来 N 个发送分组的编号

3. TCP RENO 发送 0 丢失后，估算下一次重发的时间，写出两个编号

HTML 文件，大小为 10kBYTEs；还有 10 个 1kBYTEs 的图片。链路带宽为 10Mbps。

1. 非持久 HTTP 响应时间

2. 非持久、5 个并行 TCP 响应时间

3. 带流水线

**综合题**

（2018期末）

在滑动窗口协议中，序号位数为 n 位，问当接收窗口大小分别为 1 和𝑾𝒓时，发送窗口的最大大小为多少？给出结论并证明之。

A、B双方已经建立了TCP连接，初始阈值为32K字节(1K = 1024)，最大发送段长MSS为1K字节。发送方向为A->B, B没有数据要发送, B每收到一个数据段都会发出一个应答段。在整个过程中上层一直有数据要发送, 并且都以MSS大小的段发送。A的发送序列号从0开始。

1.在传输过程中, A收到1个ACK为10240的数据段，收到这个应答段后, A处拥塞窗口的大小是多少？

2. 当收到ACK为32768的数据段后，A处拥塞窗口的大小是多少？

3. 当阈值为32K字节、拥塞窗口为40K字节时, 发送方发生了超时，求超时发生后拥塞窗口的大小和阈值的大小。

**应用层**

**选择**

（2018期末）

SMTP 的功能

SNMP 协议中哪个是服务器，哪个是客户端

（2012期末）

HTTP1.0 协议是：

a. 非坚持，得到一个对象需要一个 RTT

b. 非坚持，得到一个对象需要 两个 RTT

c. 坚持，得到一个对象需要一个 RTT

d. 坚持，得到一个对象需要两个 RTT

对于 EMAIL 下列说法正确的是：

a. 收发均使用 SMTP 协议

b. 发送使用 SMTP 协议，接收使用 POP3/IMAP 协议

c. 发送使用 POP3 协议，接收使用 SMTP 协议

d. 发送和接收均使用 POP3 协议

**简答**

（2013期末）

主机访问 www.sina.com 所需用到的协议

**综合题**

（2018期末）

给一个 URL：<http://info.tsinghua.edu.cn:80/index.jsp>

1、说出这个 URL 各个组成部分

2、一般来说，在浏览器里输入 <http://info.tsinghua.edu.cn:80/index.jsp> 跟输入

<http://166.111.4.98:80/index.jsp> 看到的是一样的。

（1）如果输前者能打开，后者打不开，这可能是什么原因？

（2）如果输前者打不开，后者能打开，这可能是什么原因？