# 数据结构：

**1.判断(10*2')：**

1)T(n)=a，无论常数 a 多大，时间复杂度为 T(N)=T(n/2)+O(1)的解总是 O(logn)
2)基于 CBA 的算法对所有大小为 n 的数组时间复杂度是Ω(nlongn)
3)基数排序的底层排序算法一定是稳定的
4)输入随机的情况下完全二叉堆的插入平均时间是常数
5)伸展树插入操作的分摊时间复杂度 O(logn)
6)对长度为 m=4k+3 素数的散列表双平方探测一定能访问其全部元素
7)没改进的 next 算法时间复杂度也是 O(n)
8)Fib 查找时以前后黄金分割点作为轴点的常系数相同
9)PFC(最优前缀编码)互换不同深度节点位置一定会破坏其性质
10)任何情况下折半查找都比顺序查找快

**2.选择(8*3')：**

1)就地算法的空间复杂度是 A.O(1) B. C. D.
2)后缀表达式扣去一个符号来猜扣去的是什么，跟去年的类似
3)一个非法表达式，问强行求解的值是多少
4)7 阶 B-树根节点常驻内存，则对规模为 2017 的 B-树最多需要几次访问？
A. B. C. D.
5)散列长为 2017，采用单平方探测，已经存入 1000 个元素，问此时最多有()个懒惰删除的桶单元
A.8 B.9 C. D.
6)分别按照递增和递减的顺序依次向平衡二叉树插入元素，则存在常数 k 使 n=2^k-1 是二者生成的平衡二叉树相等的
A.充要条件 B.必要不充分条件 C.充分不必要条件 D.不充分不必要条件
7)左式堆最右侧链长度为 k，则左式堆__含有__个元素。
A.最少 2^k B.最少 2^k-1 C.最多 ** D.最多 **
8)gs[0]=1 的概率是
A.1/m B.1/2^(m-1) C.1/2^m D.1/2^(m+1)

**3.单峰向量(13')**

已知 A[0,n ), A[0~k)严格单调递增，A[k~n)严格单调递减，设计一个 O(logn)算法找出 k
1)伪代码描述算法
2)说明算法正确性
3)证明最坏情况下时间复杂度也是 O(logn)

**4.最大和区间(13')**

给定一个整数序列，求出连续子序列和的最大值
1)说明算法思路
2)伪代码描述算法
3)说明时间复杂度和空间复杂度
题注(大致意思)：蛮力算法就不要用啦，是 O(n^3),只有设计出 O(n)算法才有可能满分，O(n^2)酌情给分。

# 计算机原理：

**1.判断**

1)提高 cpu 主频可以加快程序执行速度

2)raid6 坏两个磁盘也可以工作

3)c 语言若 int x,y 若 x>y，则-x<-y

**2.填空**

1)-2017 的 32 位补码表示__(16 进制或 2 进制)。

2)-2017 的 IEEE 单精度浮点表示__。

3)高速缓存器的几种映射方式__、__、__。

4)处理机__逻辑电路进行算术运算，__逻辑电路用于数据暂存，__逻辑电路用于分支选择。

**3.选择**

1)以下关于五段流水线的处理机说法错误的是

A.多个处理器不会发生结构冲突

B.每个周期执行一个功能

C.可以采用微程序或者硬连线设计

D.不同的指令执行时间相同

2)以下说法正确的是

A.缓存越大程序执行速度越快

B.TLB 也是一种缓存数据和指令的缓存器

C.

D.

3)以下哪个不是响应异常的处理 A.保存 pc B.保存通用寄存器 C.保存异常原因 D.恢复 pc

4)以下哪种不可以解决数据冲突 A.暂停流水线 B.分支预测 C.调整指令顺序 D.数据旁路

4.五段流水线，每段 10ns，每个寄存器 5ns，以下一段程序(4 句)，问执行时间是多少

lw ***

sub ***

and ***

or ***

# 计算机操作系统：

**1.填空(10*1')**

1)子进程执行 exit()，若未检测到父进程执行 wait()，则子进程进入__状态。

2)当某子进程调用 exit()时唤醒父进程，将 exit()返回值作为父进程中 wait()的返回值

2)高响应比调度算法的分子是__，分母是__。

3)优先级反置指的是__抢占了__的资源，__时低优先级进程能动态改变优先级

4)__支持暂时放弃互斥资源访问权，等待信号

**2.判断**

1)管程就是一个黑箱子，程序员往里面扔函数，同一时间只有一个函数在执行
2)Buddy 算法中，释放一个空间后可以根据起始长度和大小与相邻空闲空间合并
3)如果用户强制使用任务管理器 kill 一个进程，那么即使它处于就绪状态/阻塞状态，操作系统也要把它变成运行状态
4)操作系统采用 copy on write 机制时，fork()函数会复制进程的页目录表
5)使用自旋锁不能保证进程按先来后到的顺序使用 cpu 资源
6)管程和信号量在功能上等价
7)管程将资源抽象成条件变量，通过变量值的增减来控制进程的访问

**3**.LRU、BEST、CLOCK、FIFO 页面置换算法是否能产生 belady 异常，若可以举出例子，不可以给出证明(6')

**4.ucore(6')**

```
/* *
 * struct Page - Page descriptor structures. Each Page describes one
 * physical page. In kern/mm/pmm.h, you can find lots of useful functions
 * that convert Page to other data types, such as phyical address.
 * */
struct Page {
    int ref;                        // page frame's reference counter
    uint32_t flags;                 // array of flags that describe the status of the page frame
    unsigned int property;          // used in buddy system, stores the order (the X in 2^X) of the continuous memory block
    int zone_num;                   // used in buddy system, the No. of zone which the page belongs to
    list_entry_t page_link;         // free list link
    list_entry_t pra_page_link;     // used for pra (page replace algorithm)
    uintptr_t pra_vaddr;            // used for pra (page replace algorithm)
};
```

```
// convert list entry to page
#define le2page(le, member)                 \
    to_struct((le), struct Page, member)
```

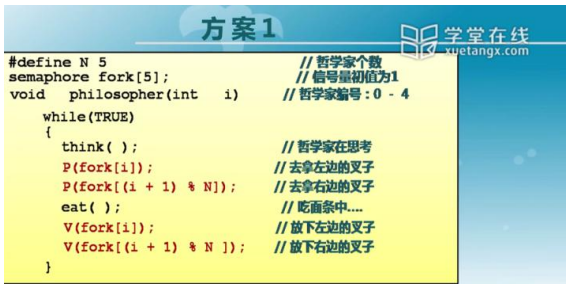
```

/* *
 * to_struct - get the struct from a ptr
 * @ptr:    a struct pointer of member
 * @type:   the type of the struct this is embedded in
 * @member: the name of the member within the struct
 * */
#define to_struct(ptr, type, member)                               \
    ((type *)((char *)(ptr) - offsetof(type, member)))
```

```
struct list_entry {
    struct list_entry *prev, *next;
};

typedef struct list_entry list_entry_t;

static inline void list_init(list_entry_t *elm) __attribute__((always_inline));
static inline void list_add(list_entry_t *listelm, list_entry_t *elm) __attribute__((always_inline));
static inline void list_add_before(list_entry_t *listelm, list_entry_t *elm) __attribute__((always_inline));
static inline void list_add_after(list_entry_t *listelm, list_entry_t *elm) __attribute__((always_inline));
static inline void list_del(list_entry_t *listelm) __attribute__((always_inline));
static inline void list_del_init(list_entry_t *listelm) __attribute__((always_inline));
static inline bool list_empty(list_entry_t *list) __attribute__((always_inline));
static inline list_entry_t *list_next(list_entry_t *listelm) __attribute__((always_inline));
static inline list_entry_t *list_prev(list_entry_t *listelm) __attribute__((always_inline));

static inline void __list_add(list_entry_t *elm, list_entry_t *prev, list_entry_t *next) __attribute__((always_inline));
static inline void __list_del(list_entry_t *prev, list_entry_t *next) __attribute__((always_inline));
```

le2page(*page,page_link)语句都需要展开那些宏定义？说明这个语句的含义。(还有一段 ucore 代码是 buddy 算法的页面分配函数，好像跟这道题关系不大，就不贴了~~主要是没找到~~。)

**5**.页面 4kB，页表项 32bit，最大能支持 4GB 的内存空间，现在有一种新技术能支持 64GB 空间，这时页表项变成 64bit，重新设计页表结构

**6.哲学家就餐问题**




1)上述算法会不会死锁，如果会请举例 2)算法是否允许两个哲学家同时进餐，若可以请举例

# 计算机网络：

**1.选择**

1)TCP/IP 与 OSI

A.

B.OSI 从上到下依次是应用层，会话层，表示层，网际层，网络层，数据链路层，物理层

C.TCP/IP 从上到下依次是应用层，网络层，数据链路层，物理层

D.TCP/IP 适用场合比 OSI 更广

2)奈奎斯特定理适用于以下哪些场合

i.同轴电缆 ii.光纤 iii.红外线

A. B. C. D. (以上三种介质的排列组合)

3)两地相距 3000 公里(传播速度 6ms/公里) 最大帧 64 字节，采用 GBN 协议，带宽为 1.544Mbps，则若要最大限度发挥网络带宽，至少需要多少比特的序号

A. B. C. D.

4)选择重传协议，序号为 0-7，发送窗口为 7，当数据发送不产生冲突是，接收窗口最大值为多少

A.4 B.5 C.7 D.8

5)dns 相关问题

A.天猫双 12 购物，不同地方两个人访问淘宝得到的 ip 一定相同

B.

C.存储 ip 是五元组

D.数据库集中存储

**2**.一道透明网桥的大题，两个网桥三段子网，建立转发表，要求填表。

**3**.一道路由器大题，两个路由器，三段网络的最大帧长度分别为 1024,512,912，报头长度分别为 14,12,12(数据不一定准确)。

拓扑结构：A—R1—R2—B（R1、R2 的 e0 接口分别连 A、B，e1 接口互连），

1） 分配给这个网络一个 192.166.1.0/24 的 ip，划分子网，使 A，B 子网中主机数量尽可能多，写出子网以及 R1，R2 的 e0、e1 接口的 ip 地址和子网掩码

2） 现在要发送一个长度为 900B 的 tcp 数据段，tcp 首部 20B，ip 首部 20B，identification 值为 X，问这个数据段经过 A-R1、R1-R2、R2-B 三段子网的时候 total_length、identification、DF、MF、offset 值分别为多少

3） A 要给 B 发 7 个数据段，建立了一个 TCP 连接，已知往返时间为 RTT，问从建立连接开

始到发送结束共持续了多长时间