辽宁大学考研专业课 854(专硕)真题答案

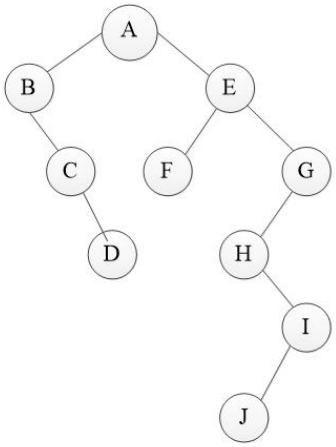
（2013 年—2018 年）

（2013 年）

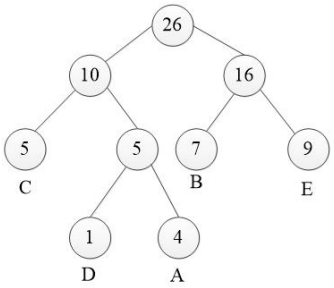
一、BDCCB　AADBA　　BDBDD

二、1.

中序遍历：BCDAFEHJIG



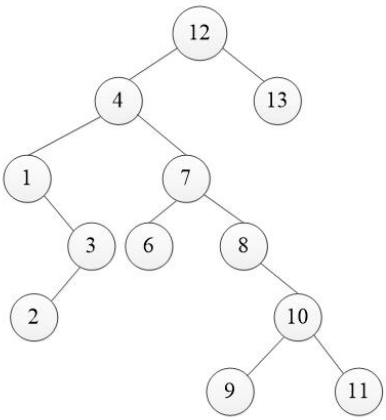
2.（1）



（2）$WPL=(5+7+9)*2+(1+4)*3=57$

（3）$A:011;B:10;C:00;D:010;E:11.$

3.



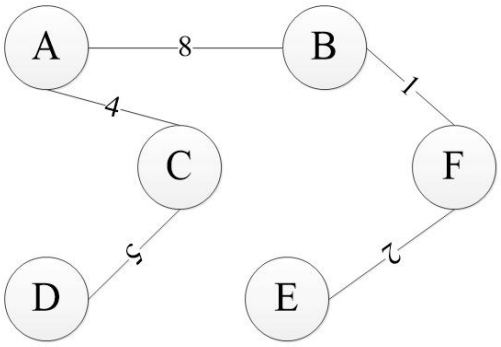
$ASL=(1+2*2+3*2+4*3+5*2+6*2)/12=15/4$

4. 输出二叉树中权值大于等于 100 小于等于 999 的结点数。

5.（1）从栈顶开始，依次是 1,2,6,9；

（2）将栈中元素倒序排列并删除指定元素 e.

6.最小生成树如下：

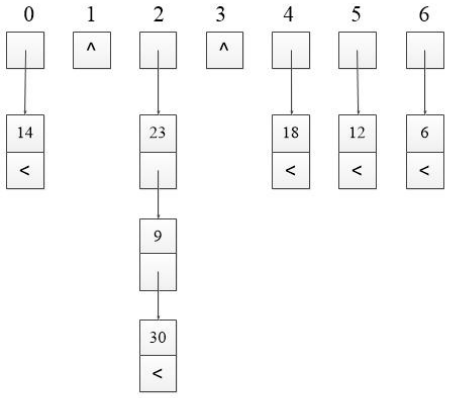


生成边次序：BF、FE、AC、CD、BA

7. 哈希地址如下表：

散列地址	0	1	2	3	4	5	6
关键字	14	18	23	9	30	12	6

则由此构造的链地址法处理冲突的 Hash 表如下：



$ASL_{成功}=(1*5+2*1+3*1)/7=10/7$ （这里的 7 是比较次数。）

注：$ASL_{失败}=(1+0+3+0+1+1+1)/7=1$ （这里的 7 是散列后的地址个数。）

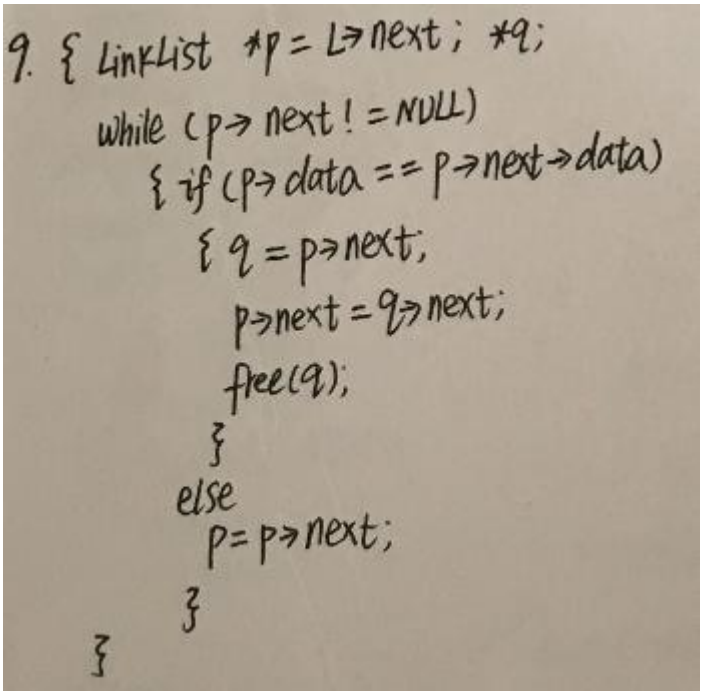
8.（1）①在图中选择一个没有前驱（入度为 0）的顶点数；

②删除①中的顶点，并且删除以该顶点发出的全部边；

③重复上述两步，直到剩余的网中不存在没有前驱的顶点为止。

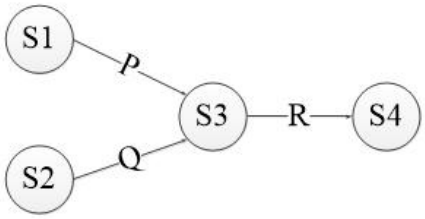
（2）①1,5,2,6,3,7,4,8；②1,2,3,4,5,6,7,8

9.



```
9. { LinkList *p = L->next; *q;
     while (p->next != NULL)
       { if (p->data == p->next->data)
           { q = p->next;
             p->next = q->next;
             free(q);
           }
         else
           p = p->next;
       }
   }
```

10.




```
begin parbegin
        begin S1;signal(P);end
        begin S2;signal(Q);end
        begin wait(P);wait(Q);S3;signal(R);end
        begin wait(R);S4;end
end parend
```

11.解：①40K=1024*40=40960,地址为 40992；

②段长 3K,即 3072,4200>3072,发生越界；

③6 不在段表内，发生缺段中断。

12.解：①首次适应算法：按分区号递增的次序查找，14K 分配到 3 分区，12K 分配到 1 分区，17K 分配到 4 分区，20K 请求分配资源失败；

②最佳适应算法：将空闲区按容量大小递增次序排列

分区长	6K	9K	11K	12K	14K	17K	20K	22K
分区号	2	5	6	1	7	8	4	3

则 14K.12K,17K.20K 被分别分配到 7,1,8,4 分区。

13. 解：P3 请求资源后，Allocation=(0,6,4,2,3),Need=(0,0,4,2,0),Available=(1,0,1,2,2).

①P3 请求 Request(0,6,1,0,2)后，P3 还需资源(0,0,4,2,0),系统剩余资源数(1,0,1,2,2);

②此时剩余资源满足 P0 还需要资源数，可用资源数(1,0,4,4,3);
③满足 P3 还需资源，回收资源后资源数为(1,6,8,6,6);
④此时满足 P4，回收资源后资源数为(1,6,9,10,6);
⑤此时满足 P1，回收 P1 资源后，资源数(2,6,9,10,6);
⑥此时满足 P2，回收资源后，资源数(3,9,14,14,8);

此时存在安全序列 $P_0 \rightarrow P_3 \rightarrow P_4 \rightarrow P_1 \rightarrow P_2$

或：

P	Work	Need	Available	Work+Available	Finish
	A B C D E	A B C D E	A B C D E	A B C D E	
P0	1 0 1 2 2	0 0 1 2 1	0 0 3 2 1	1 0 4 4 3	T
P3	1 0 4 4 3	0 0 4 2 0	0 6 4 2 3	1 6 8 6 6	T
P1	1 6 8 6 6	1 6 5 0 0	1 0 0 0 0	2 6 8 6 6	T
P2	2 6 8 6 6	2 3 5 6 1	1 3 5 4 2	3 9 13 10 8	T
P4	3 9 13 10 8	0 6 5 6 0	0 0 1 4 0	3 9 14 14 8	T

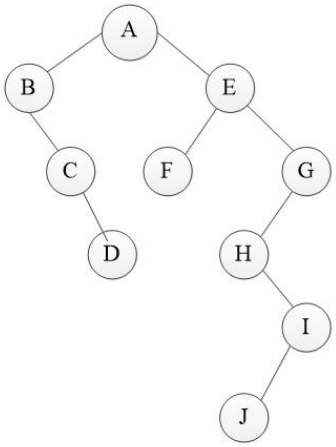
14.解：（1）
$$10KB+1KB*1/4K+1KB*1/4K*1/4K+1KB*1/4K*1/4K*1/4K$$
$$=10KB+1/4MB+1/16GB+1/64TB$$

（2）29848/256=116MOD152，前 10 项占 10 个盘块，则可以从索引节点内第 10 个地址项，并从一次间接地址块的第 106 项中获取对应物理块号，块内位移 152；
（3）读取文件的索引结点到内存中（访问一次），这个文件的页挂在三级索引下，读 3 个索引块需要访问磁盘 3 次（已访问磁盘 4 次），得到该页的物理地址，再去读这个页（已访问磁盘 5 次），因此，磁盘最多启动 5 次。

（2014 年）

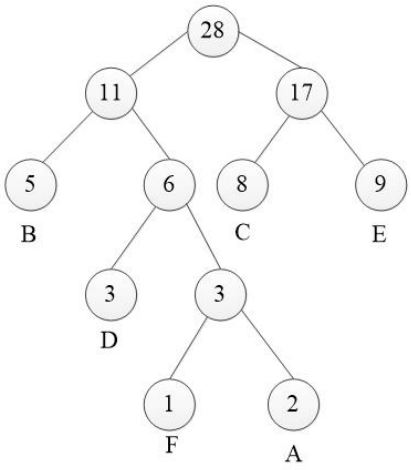
一、BDCCB　BDCAB　　ABBCB

二、1.



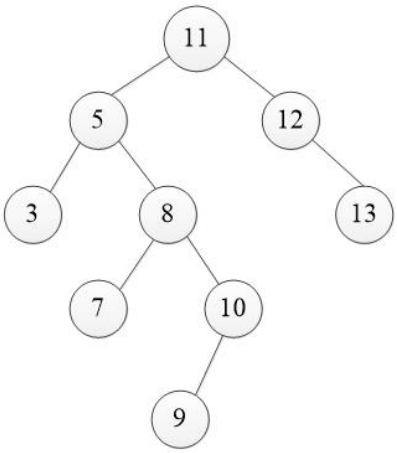
后序遍历：DCBFJIHGEA

2.



$$WPL=(5+8+9)*2+3*3+(1+2)*4=65$$

3.



$$ASL_{成功}=(1+2*2+3*3+2*4+1*5)/9=3$$

4.（1）（5,5）；（2）输出元素 e

5.un(3):value=un(2)+8;

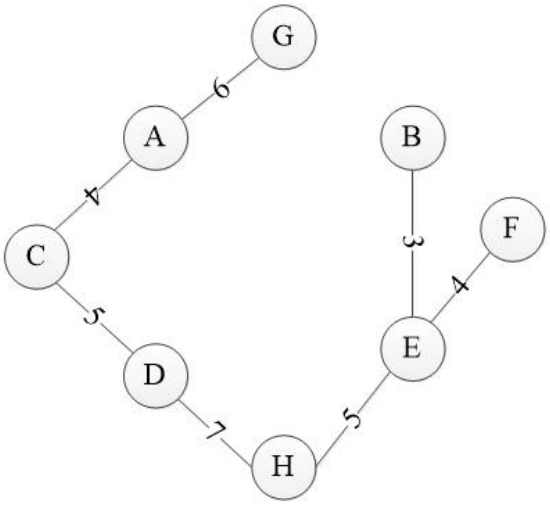
un(2):value=un(1)+8;

un(1):value=un(0)+8;

un(0)=3;

则 un(3)=27.

6.生成边次序：AC　CD　DH　HE　EB　EF　AG



7. $ASL_{成功}=(1+2+1+4+3+1+1+3+1+1+3+2)/12=23/12$

散列地址	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
关键字		01	14	55	27	68	19	20	84		23	11	10	77		
比较次数		1	2	1	4	3	1	1	3		1	1	3	2		

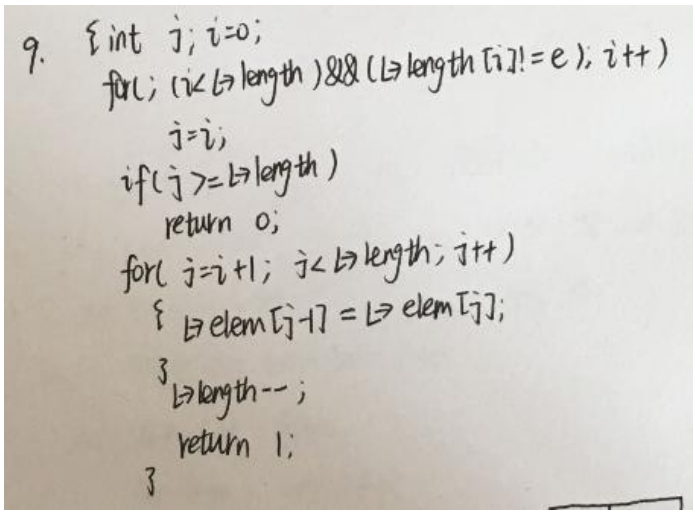
8.（1）①在图中选择一个没有前驱（入度为 0）的顶点数；

②删除①中的顶点，并且删除以该顶点发出的全部边；

③重复上述两步，直到剩余的网中不存在没有前驱的顶点为止。

（2）①H,A,F,G,D,E,B；②H,F,A,G,E,D,B

9.



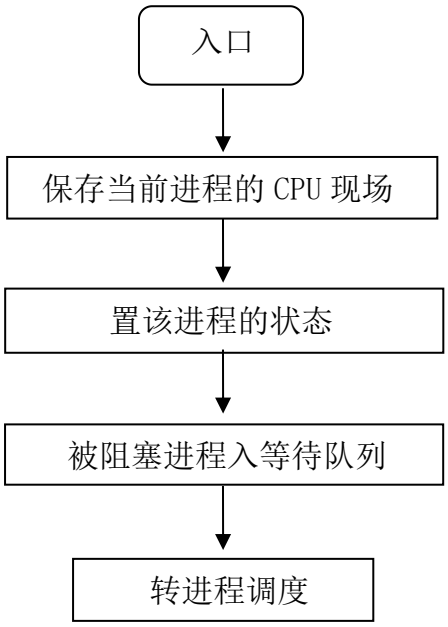
10.（1） $N=8KB/4B=2K$,最大文件长度 $2K*8KB=16MB$

（2）若采用二级索引，最大文件长度可达 $N*N*8KB=32GB$

11.

过程：首先中断 PCB，停止进程运行，将 CPU 的现行状态存放到 PCB 的 CPU 状态保护中，然后将该进程置阻塞状态，并把它插入等待队列中，然后系统执行调度程序，将 CPU 分配给另一个就绪的进程。

阻塞原语流程图：



阻塞原语算法：

```
void block(void)
{
        i=EP;
        stop(i);                    /*阻塞调用进程自己*/
        i.status="阻塞";          /*设置阻塞状态*/
        i.state=WQ（r）;          /*填写阻塞队列名称*/
        insert(WQ(r),i);          /*把调用进程的 PCB 插入相应等待队列 WQ（r）*/
        scheduler;                  /*转进程调度程序重新调度*/
}
```

12.

```
Var mutex, empty, full: semaphore:=1, 1, 0;
DataCollection:                          DataCompute:
begin                                    begin
    repeat                                   repeat
```

```
……                                  ……
gather data in nextp;               wait(full);
wait(empty);                        wait(mutex);
wait(mutex);                        nextc:=B;
B:=nextp;                           signal(mutex);
signal(mutex);                      signal(empty);
signal(full);                       compute data in nextc;
until false;                    until false;
end                                 end
```

13.答：对于程序 1，首次缺页中断（访问 A[0，0]时产生）将装入数据的第 1、2 行共 200 个整数，由于程序是按行对数组进行访问的，只有在处理完 200 个整数后才会再次产生缺页中断；以后每调入一页，也能处理 200 个整数，因此处理 100×100 个整数共将发生 50 次缺页。

对于程序 2，首次缺页中断（访问 A[0，0]时产生）将装入数据的第 1、2 行共 200 个整数，但由于程序是按列对数组进行访问的，因此在处理完 2 个整数后又会再次产生缺页中断；以后每调入一页，也只能处理 2 个整数，因此处理 100×100 个整数共将发生 5000 次缺页。

14.64 个页面相当于 2^6，也就是 6 位表示页，，1KB 相当于 2^{10}，也就是 10 位表示块号。

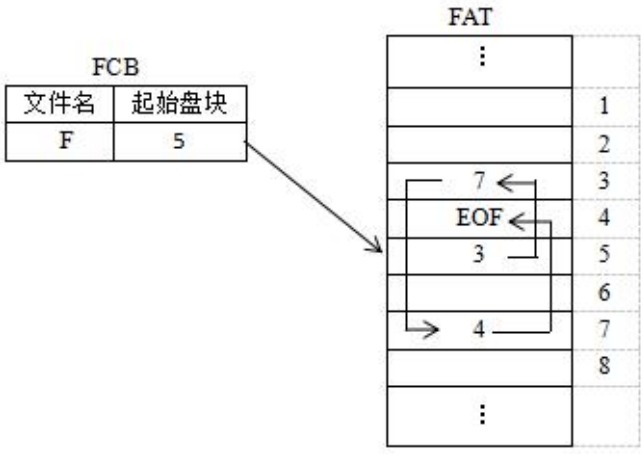
（1）0A5C 二进制是 0000 1010 0101 1100，000010 是 2,2 分配的物理块号是 4，即 000100，物理地址：0001 0010 0101 1100，化为十六进制 125C(H);

（2）103C 二进制是 0001 0000 0011 1100，000100 是 4，分配的空闲物理块号是 6，即 000110，物理地址：0001 1000 0011 1100，化为十六进制 183C(H).

15.（1）$\left(12*2^{10}*2^{10}\right)KB/4KB=3*2^{20}$，FAT 表项长度为字节偶数倍，则表项取 32 位，

FAT 表至少需占用 $32/8*3*2^{20}=12MB$。

（2）



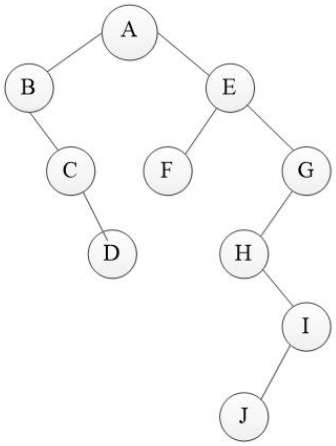
16. 首次适应算法：分别分配到分区 3,1,4,18K 分配失败；

最佳适应算法：分别分配到分区 7,1,8,4.

（2015 年）

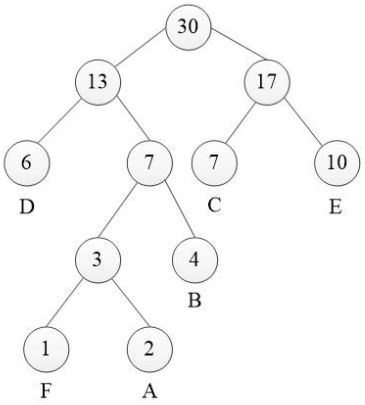
一、BDCCB　BDCAC　ABBDB

二、1.

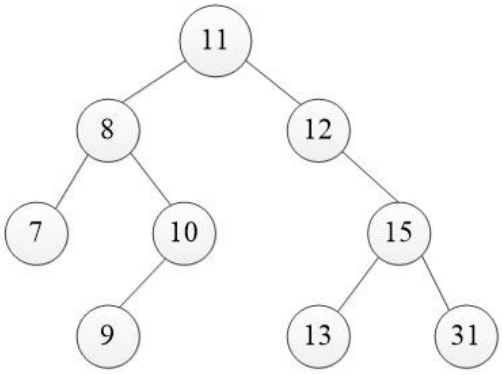


后序遍历：DCBFJIHGEA

2. $WPL=(6+7+10)*2+4*3+(1+2)*4=70$



3.

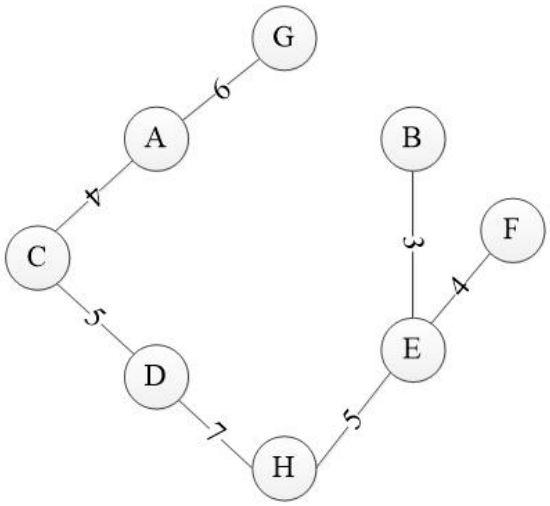


$ASL=(1*1+2*2+3*3+4*3)/9=26/9$

4.un(3)=un(2)+8=27;
un(2)=un(1)+8=19;
un(1)=un(0)+8=11;
un(0)=3.

5.（1）（5,5）；（2）输出元素 e

6.生成边次序：FE　EB　EH　HD　DC　CA　AG



7. $ASL_{成功}=(1+1+1+1+2+1+1+1+1+3+1)=14/11$

散列地址	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
关键字	48		18	19	52	20		23			10	27	12	11	14	
比较次数	1		1	1	1	2		1			1	1	1	3	1	

8.（1）①在图中选择一个没有前驱（入度为 0）的顶点数；

②删除①中的顶点，并且删除以该顶点发出的全部边；

③重复上述两步，直到剩余的网中不存在没有前驱的顶点为止。

（2）①H,A,F,D,G,B,E,C；②H,F,A,D,G,E,B,C

9.Delete(SqList *L,int e){
int *p,*q;
while(p!=null);{
if(p->data=e)
p=q;
{p=p->next;
delete q;
}
else p=p->next;
q=q->next;
return 1;
}
else return 0;
}

10.（1）$\lceil 1000/32 \rceil = 32$　（2）$32*i+j$

11.（1）$2^8=256$ 段；（2）$2^{16}=64KB$；

（3）$[2 \quad 80]$：所在第 2 段没有在内存中，无法进行地址变换，发生缺页中断；

$[3 \quad 50]$：主存地址：4000+50=4050

12.64 个页面相当于 2^6，也就是 6 位表示页，，1KB 相当于 2^{10}，也就是 10 位表示块号。

（1）0B53 二进制是 0000 1011 0101 0011，000010 是 2,2 分配的物理块号是 7，即 000111，物理地址：0001 1111 0101 0011，化为十六进制 1F53(H);
（2）04AB 二进制是 0000 0100 1010 1011，000001 是 1，1<8,属于合理页面，分配的空闲物理块号是 8，即 001000，物理地址：0010 0000 1010 1011，化为十六进制 20AB(H).

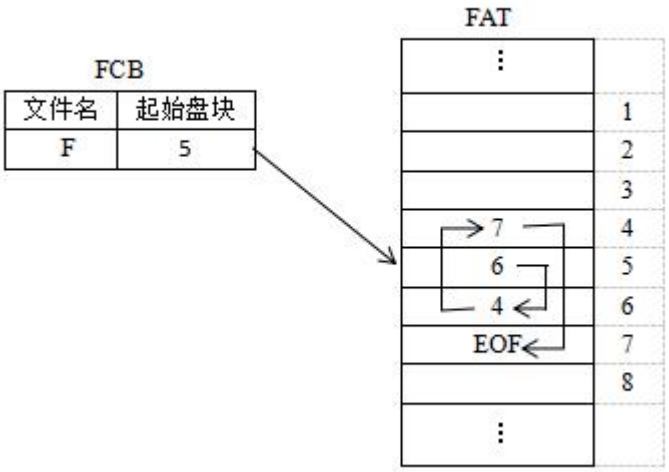
13.（1）90,98,100,80,70,46,38,10,125,135,160;

（2）$(8+2+20+10+24+8+28+115+10+25)/10=25$

14.（1）$(16*2^{10}*2^{10})KB/4KB=4*2^{20}$ 块，取 32 位表项长度，则字节占用

$32/4*4*2^{20}=16MB$

（2）



15.semaphore empty=100；
semaphore full=0;
semaphore s=1;

process Pin()
begin
L1:生产了一台设备；
P(empty);
P(S);
使用运输汽车入库；

process Pout()
begin
L2:P(full);
P(S);
使用运输车出库；
V(S);

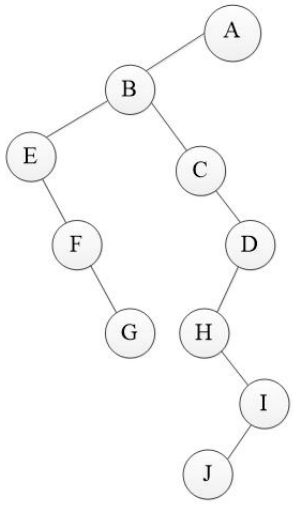
V(S);
V(full);
go to L1;
end;

V(empty);
提出设备供应客户；
go to L2;
end;

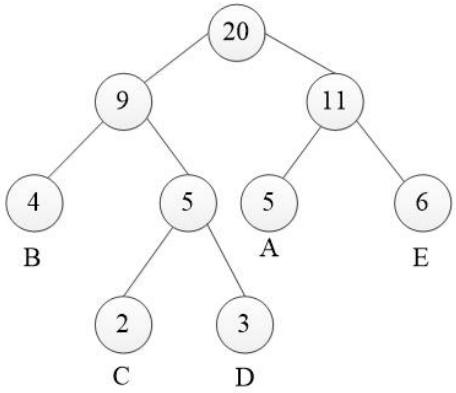
（2016 年）

一、ABACC　BACCD　CCCDC

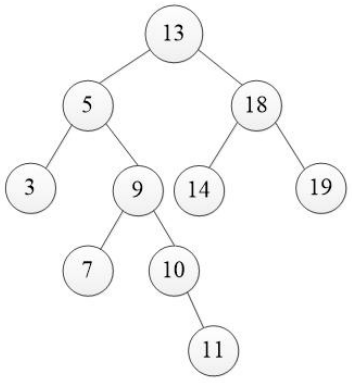
二、1.中序遍历：EFGBCHJIDA



2. $WPL=(4+5+6)*2+(2+3)*3=45$



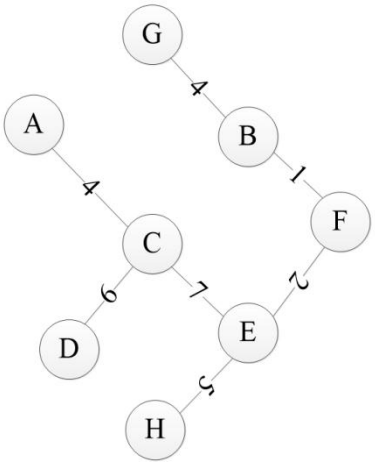
3. $ASL=(1*1+2*2+4*3+2*4+1*5)/10=3$



4.un(3)=un(2)+8=27;
un(2)=un(1)+8=19;
un(1)=un(0)+8=11;
un(0)=3.

5.（1）(5,5)；（2）输出元素 e

6.生成边次序：AC　CD　CE　EF　FB　GB　EH



7. $ASL_{成功}=(1+1+1+1+3+1+2)/7=10/7$

散列地址	0	1	2	3	4	5	6	7	8	9
关键字	7	15	2	31	23	19	12			
比较次数	1	1	1	1	3	1	2			

8. （1）①在图中选择一个没有前驱（入度为 0）的顶点数；
②删除①中的顶点，并且删除以该顶点发出的全部边；
③重复上述两步，直到剩余的网中不存在没有前驱的顶点为止。
（2）①H,A,F,G,D,E,B,；②H,A,F,G,E,D,B

9.
```
LinkListDelete(linkList L){
    LinkList p,q,head;
    p=(LinkList)malloc(sizeof(L));
    p->next=NULL;
    q=head;
    p=head->next;
    while(p!=NULL){
    if(p->data%2 !=0){
    p=p->next;
    q=q->next;
}else{
   p->next=q->next;
   p=p->next;
}
}
```

10.①当用户要求对一个文件实施多次读/写或其他操作时，每次都要从检索目录开始，为避免多次或重复的检索目录，OS 提供“打开”这一文件系统调用，即在用户和指定文件之间建立起一个连接，当用户再次向系统发出文件系统操作请求时，系统根据用户提供的索引号直接在文件打开表中查找文件信息；

②若用户不再需要对该文件实施相应的操作，可利用“关闭”系统调用来关闭此文件，即断开此连接，OS 将会把文件从打开文件表的表目上删除。

11.（1）$4KB/4B=1K$，$1K*4KB=4MB$；

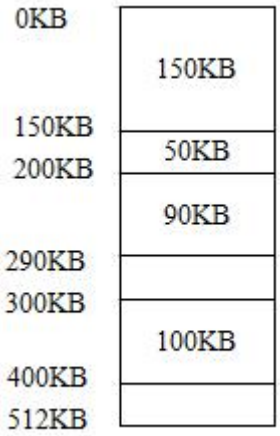
（2）$1K*1K*4KB=4GB$。

12.（1）0B61 二进制是 0000 1011 0110 0001，000010 是 2,2 分配的物理块号是 9，即 001001，物理地址：0010 0111 0110 0001，化为十六进制 2761(H);

（2）1742 转换为二进制为 0001 0111 0100 0010，000101 是 5，没有可以分配的物理块；

（3）1A63 转换为二进制为 0001 1010 0110 0011，000110 是 6，没有可以分配的物理块.

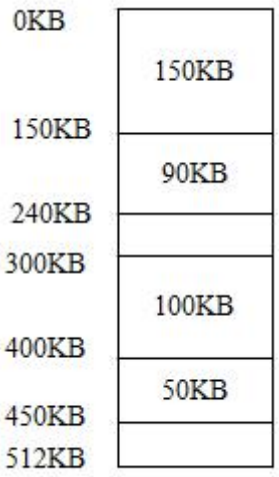
13.（1）首次适用算法：



空闲区：第一块：起始地址 290KB，大小 10KB；

第二块：起始地址 400KB，大小 112KB。

（2）最佳适用算法：

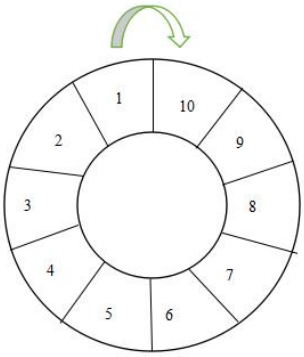


空闲区：第一块：起始地址 240KB，大小 60KB；

第二块：起始地址 450KB，大小 62KB。

（3）若随后又申请 80KB，则最先适配算法可以分配成功，而最佳适配算法没有足够大的空闲分区分配。这说明最先适配算法尽可能适用了低地址部分的空闲区域，留下了高地址部分的大的空闲区，更有可能满足进程申请。

14.



（1）读一个逻辑记录的时间 20ms/10=2ms，读出后还需要 4ms 处理时间，故当磁头处于某记录起点时，处理它需要 6ms 时间，又逻辑记录按逆时针方向安排，因此处理完一个逻辑记录后将磁头转到下一个逻辑记录需要 16ms 时间（6ms 后磁盘已经到达 4 的位置），故处理完这 10 个记录需要花费 $6+9*(16+6)=204ms$

（2）可使处理程序处理完一个记录后，磁头刚好转到下一个记录的始点，顺序为：记录 1，记录 4，记录 7，记录 10，记录 3，记录 6，记录 9，记录 2，记录 5，记录 8，共需要时间 6*10=60ms.

15.semaphore mutex=1;
semaphore countB=0;
semaphore countA=0;

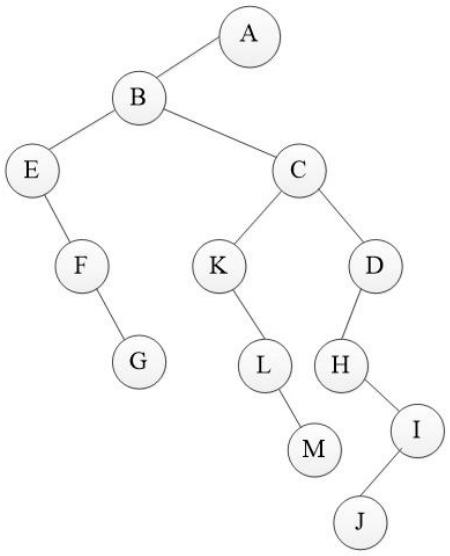
```
A(){
 while(1){
    if(countB==0)
    wait(mutex);
    if(countB<=30)
    进入房间；
    countA++;
    离开房间；
    countA--;
    if(countA==0)
    signal(mutex);
}
}
```

```
B(){
 while(1){
    if(countA==0)
    wait(mutex);
    if(countB<=30)
    进入房间；
    countB++;
    离开房间；
    countB--;
    if(countB==0)
    signal(mutex);
}
}
```

（2017 年）

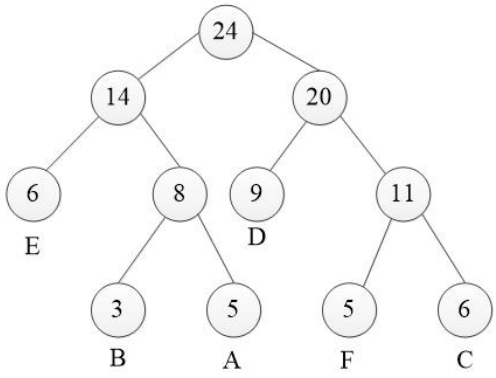
一、BCCCA　ACCAB　DDAAD

二、1.

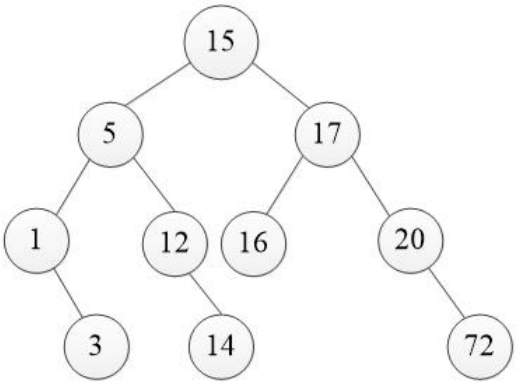


先序遍历：ABEFGCKLMDHIJ;
中序遍历：EFGBKLMCHJIDA;
后序遍历：GFEMLKJIHDCBA.

2. $WPL=(6+9)*2+(3+5+5+6)*3=87$



3. $ASL=(1*1+2*2+3*4+4*3)/10=2.9$

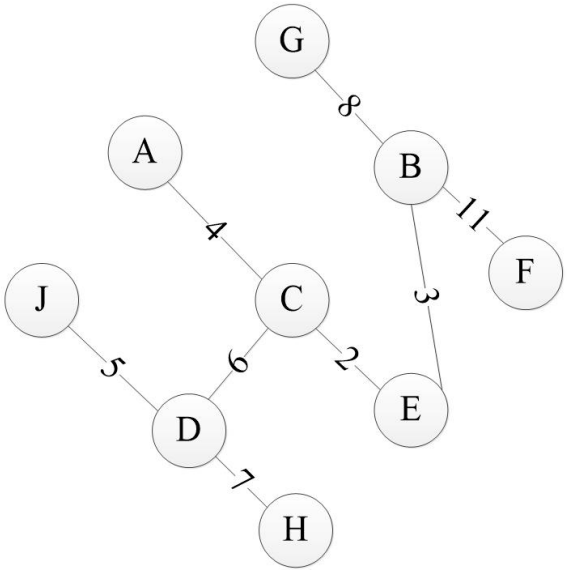


4.un(5)=un(4)+un(3)=18+11=29;
un(4)=un(3)+un(2)=11+7=18;
un(3)=un(2)+un(1)=7+4=11;
un(2)=un(1)+un(0)=4+3=7;
un(1)=4;un(0)=3.

5.（1）$(6,5,4,9)$；

（1）demo 函数的功能：从栈顶取出元素，值是否等于 e，若不等于 e，则进队，再将队列中元素按顺序存入栈中。假设栈中 n 个远古三，则当栈中元素全部出去时，执行 n 次，若有 m 个元素等于 e，则进栈元素 n-m 次，时间复杂度 O(n).

6.生成边次序：CE　EB　CA　CD　DJ　DH　BG　BF

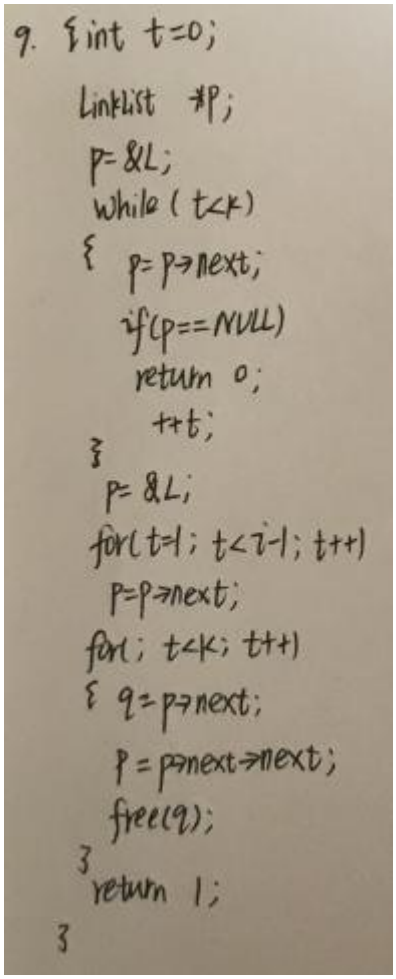


7. $ASL_{成功}=(1+1+3+3+1+1+1)/7=11/7$

散列地址	0	1	2	3	4	5	6	7	8
关键字		10	2	19	11	23		7	35
比较次数		1	1	3	3	1		1	1

8.（1）①在图中选择一个没有前驱（入度为 0）的顶点数；
②删除①中的顶点，并且删除以该顶点发出的全部边；
③重复上述两步，直到剩余的网中不存在没有前驱的顶点为止。
（2）①H,A,J,F,G,D,E,B,；②H,J,A,F,G,E,D,B

9.



```
9. {int t=0;
   LinkList *p;
   p=&L;
   while(t<k)
   { p=p->next;
     if(p==NULL)
     return 0;
      ++t;
   }
    p=&L;
   for(t=1; t<i-1; t++)
    p=p->next;
   for(; t<k; t++)
   { q=p->next;
    p=p->next->next;
    free(q);
   }
   return 1;
}
```

10.答：①调度：线程是独立调度的单位；

②不拥有资源：线程不拥有西永西苑（除必不可少的资源外），但线程可访问其隶属进程；

③并发性：同一进程内的多个进程之间也可以并发执行；

④系统开销：线程切换时，只需保存和设置少量寄存器内容，开销很小。

11.表格如下：

作业名	提交时间	执行时间	开始时间	完成时间	周转时间	带权周转时间
J1	10.0	2.0	10.0	12.0	2	1
J2	10.2	1.0	12.0	13.	2.8	2.8
J3	10.4	0.5	13.0	13.5	3.1	6.2
J4	10.5	0.3	13.5	13.8	3.3	11

平均周转时间 $=(2+2.8+3.1+3.3)/4=2.8$;

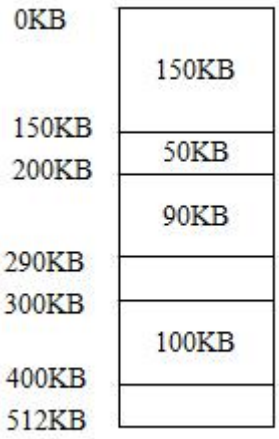
平均带权周转时间 $=(1+2.8+6.2+11)/4=5.25.$

12.（1）$7*4KB+100=28692$;

（2）在请求分页存储管理系统中，系统是通过页表进行地址变换的，先将逻辑地址分解成页号 P 和页内地址 W 两部分，然后通过查页表获得页号 P 对应的物理块号 B，将物理块

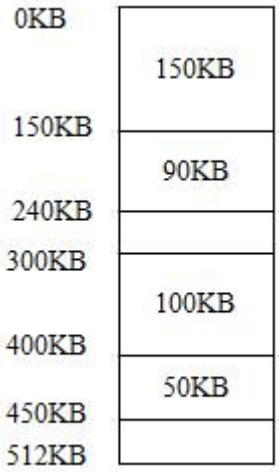
号乘以页面大小，再将页内地址 W 加到其上，即获得最终物理地址：物理地址＝块号*页大小＋页内地址。

13.（1）首次适用算法：



空闲区：第一块：起始地址 290KB，大小 10KB；
第二块：起始地址 400KB，大小 112KB。

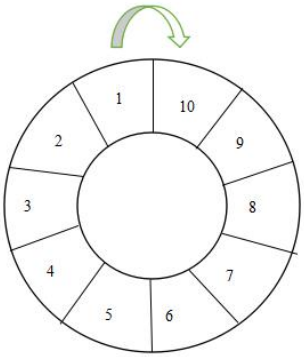
（2）最佳适用算法：



空闲区：第一块：起始地址 240KB，大小 60KB；
第二块：起始地址 450KB，大小 62KB。

（3）若随后又申请 80KB，则最先适配算法可以分配成功，而最佳适配算法没有足够大的空闲分区分配。这说明最先适配算法尽可能适用了低地址部分的空闲区域，留下了高地址部分的大的空闲区，更有可能满足进程申请。

14.



（1）读一个逻辑记录的时间 20ms/10=2ms，读出后还需要 4ms 处理时间，故当磁头处于某记录起点时，处理它需要 6ms 时间，又逻辑记录按逆时针方向安排，因此处理完一个逻辑记录后将磁头转到下一个逻辑记录需要 16ms 时间（6ms 后磁盘已经到达 4 的位置），故处理完这 10 个记录需要花费 $6+9*(16+6)=204ms$

（2）可使处理程序处理完一个记录后，磁头刚好转到下一个记录的始点，顺序为：记录 1，记录 4，记录 7，记录 10，记录 3，记录 6，记录 9，记录 2，记录 5，记录 8，
共需要时间 6*10=60ms.

15.semaphore seats=100;
semaphore readers=0;
semaphore mutex=1;

```
process getin{
 while(1){
 P(seats);//没有座位即离开
 P(mutex);
 填写登记表；
 进入阅览室读书；
 V(mutex);
 V(readers);
  }
}
```

```
process getout{
 while(1){
 P(readers);//阅览室是否有人读书
 P(mutex);
 消掉登记；
 离开阅览室；
 V(mutex);
 V(seats);
  }
}
```

（2018 年）

1、

线程是程序执行流的最小单元。一个标准的线程由线程 ID，当前指令指针(PC)，寄存器集合和堆栈组成。

进程和线程都是由操作系统程序运行的基本单元，系统利用该基本单元实现系统对应用的并发性。线程是进程中的一个实体，是被系统独立调度和分派的基本单位，线程自己不拥有系统资源，但它可与同属一个进程的其它线程共享进程所拥有的全部资源。一个线程可以创建和撤消另一个线程，同一进程中的多个线程之间可以并发执行。在有进程和线程的系统中，进程是系统资源分配的独立单位，而线程是可调度运行的独立单位。

2、

逻辑结构：指一个文件在用户面前所呈现的形式。

物理结构：指文件在文件存储器上的存储形式。

逻辑结构有两种形式：①记录式文件(有结构式文件)②字符流式文件（无结构式文件），也称流式文件

物理结构的形式：①连续文件结构②串联文件结构③索引文件结构④散列文件结构

3、

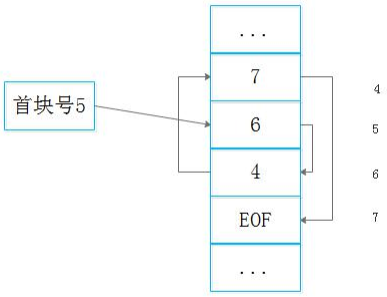
（1）主存分为 256 块，说明一共存放 256 个页面，每块大小为 4KB，所以每页大小 4*1024=4096,9016=4096*2+824，所以页号为 2，根据表中页号 2 所对应块号为 32，所以物理地址为 32*4KB+824=952

（2）同理，12300=4096*3+12，页号为 3，状态为 1，表示并未被主存装入，由此将产生缺页中断。

4、

（1）16GB/4K=4M 块 故需要 22 位对盘块进行编号，由于考虑存取方便，FAT 表每个表项长度为字节的整数倍，所以用 24 位对盘块编号，24/8=3 个字节 3*4M=12MB

（2）



5、

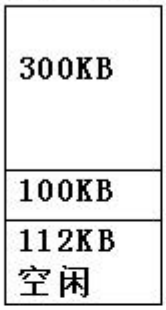
```
Var student,computer,enter,finish,test:semaphore:=0,2m,0,0,0;
student:begin
    P(computer);//得到一台计算机
    V(student);//有学生到达，通知门卫
    P(enter);//等待进入
    Practice;
    V(finish);//实习结束，通知教师
    P(test);//等待教师检查
    V(computer);//释放计算机资源
End;
Teacher:begin
    P(finish);//等待学生实习结束
    P(finish);//等待另一学生实习结束
    Check;
    V(test);//检查完成
    V(test);//检查完成
End;
Guard:begin
   P(student);//等待学生到达
   P(student);//等待另一学生到达
   V(enter);//允许学生进入
   V(enter);//允许另一学生进入
End.
```

6、

（1）首次适用算法

申请 300KB

申请 100KB



释放 300KB

申请 150KB

申请 50KB

申请 90KB

内存的空闲分区有两块

1 起始地址是 290，大小是 10KB

2 起始地址是 400，大小是 112KB

（2）最佳适用算法

申请 300KB

申请 100KB

释放 300KB

申请 150KB

申请 50KB

申请 90KB

内存的空闲分区有两块

1 起始地址是 240，大小是 60KB

2 起始地址是 450，大小是 62KB

（3）如果再申请 80KB，首次适用算法可以分配出 80KB 而最佳适用算法则没有 80KB 的连续空闲分区让其申请

首次适用算法

150KB
50KB
90KB
10KB 空闲
100KB
80KB
32KB 空闲

7、FCFS:

作业执行次序	执行时间	优先级	等待时间	周转时间	带权周转时间
1	10	3	0	10	1
2	1	1	10	11	11
3	2	3	11	13	6.5
4	1	4	13	14	14
5	5	2	14	19	3.8

系统中作业的平均周转时间为 T=(10+11+13+14+19)/5=13.4

系统中作业的平均带权周转时间为 W=(1+11+6.5+14+3.8)/5=7.26

RR：

作业	周转时间	带权周转时间
1	19	1.9
2	2	2
3	7	3.5
4	4	4
5	14	2.8

各作业在系统中的执行情况为：

（1,2,3,4,5），（1,3,5），（1,5,1,5,1,5），（1,1,1,1,1）

作业 1 的周转时间 T1=19；T2=2；T3=7;T4=44;T5=14

系统中作业的平均周转时间为 T=(19+2+7+4+14)/5=9.2

系统中作业的平均带权周转时间为 W=(1.9+2+3.5+4+2.8)/5=2.84

SJF：

作业执行次序	执行时间	优先级	等待时间	周转时间	带权周转时间
2	1	1	0	1	1
4	1	4	1	2	2
3	2	3	2	4	2
5	5	2	4	9	1.8
1	10	3	9	19	1.9

系统中作业的平均周转时间为 T=(1+2+4+9+19)/5=7

系统中作业的平均带权周转时间为 W=(1+2+2+1.8+1.9)/5=1.74

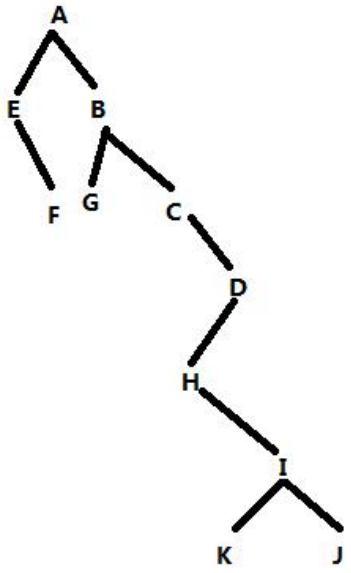
非剥夺式优先级调度算法：

作业执行次序	执行时间	优先级	等待时间	周转时间	带权周转时间
2	1	1	0	1	1
5	5	2	1	6	1.2
1	10	3	6	16	1.6
3	2	3	16	18	9
4	1	4	18	19	19

系统中作业的平均周转时间为 T=(1+6+16+18+19)/5=12

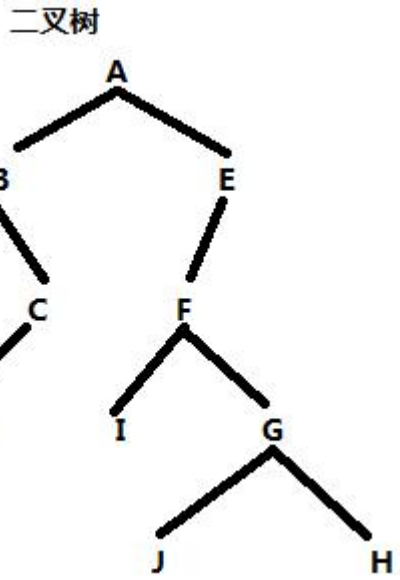
系统中作业的平均带权周转时间为 W=(1+1.2+1.6+9+19)/5=6.36

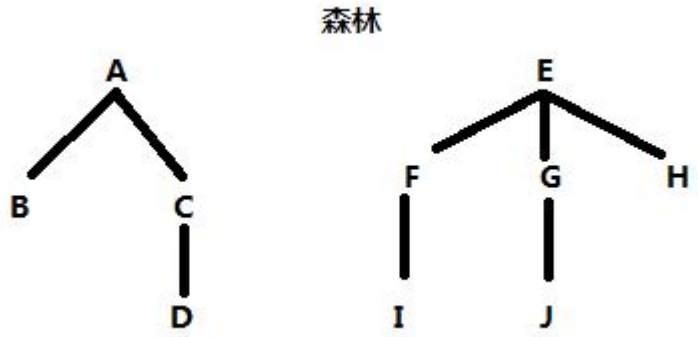
8、



9、

由于森林的先序序列和二叉树的先序序列是对应的，而森林的后序序列是和二叉树的中序序列对应的，因此根据先序 ABCDEFIGJH 和中序 BCDAIFJGHE 得到相应的二叉树，，再根据二叉树得到相应的森林





10、

（2）

A:00

B:01

C:11000

D:11001

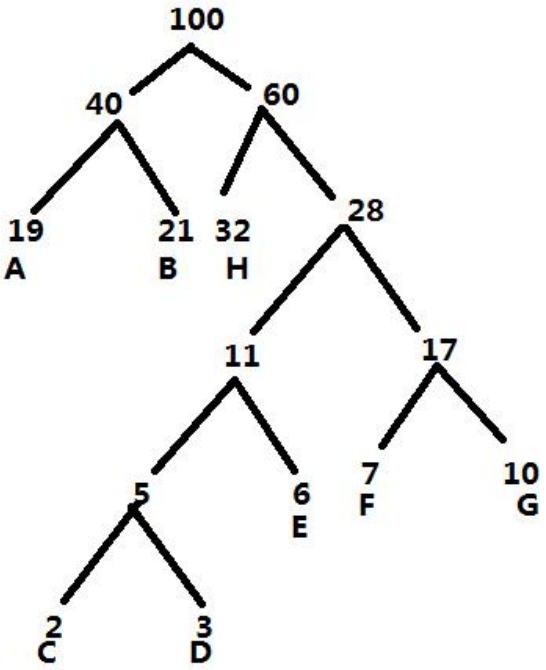
E:1101

F:1110

G:1111

H:10

(1)



11.又称普里姆算法，以图上的顶点为出发点，逐次选择到最小生成树顶点集距离最短的顶点为最小生成树的顶点，并加入到该顶点集，直到包含所有的顶点。

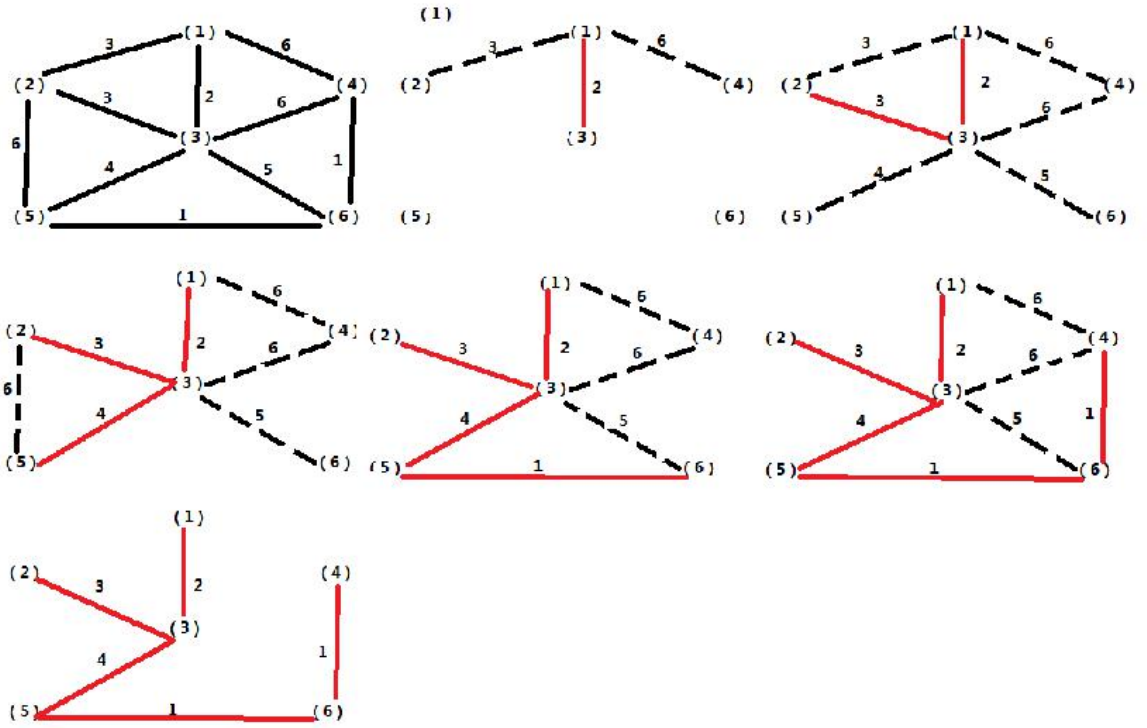
步骤：

1.选择一出发点，加入集合 A。

2.遍历与集合 A 中的点相邻的边，找到最短的边，并且不构成回路。

3.将步骤 2 得到的边的目标点加入集合 A。

4.重复 2，3 直到所有结点都加入到集合 A 中。



12、

（1）

1、选择一个入度为 0 的顶点并输出

2、从网中删除此顶点及所有出边

3、循环 1、2 直到所有顶点被去除或者无法找到入

度为 0 的顶点为止，结束后，若输出的顶点数小于网中的顶点数，则输出“有回路”信息，否则输出的顶点顺序就是拓扑序列。

AEBCFD/AEFBCD/EABCFD/EAFBCD/EFABCD

13、

13	18	24	35	47	50	62	83	90
0	1	2	3	4	5	6	7	8

Mid=0+8/2=4 62>r[4]

Mid=[(4+8)/2]=6 62=r[6]

比较次数为 2

ASL=(1*1+2*2+3*4+4*2)/9=25/9

14、

I=0 j=8 flag=54

34,23,89,48,64,50,25,90,54

34,23,54,48,64,50,25,90,89

34,23,25,48,64,50,54,90,89

34,23,25,48,54,50,64,90,89

34,23,25,48,50,54,64,90,89

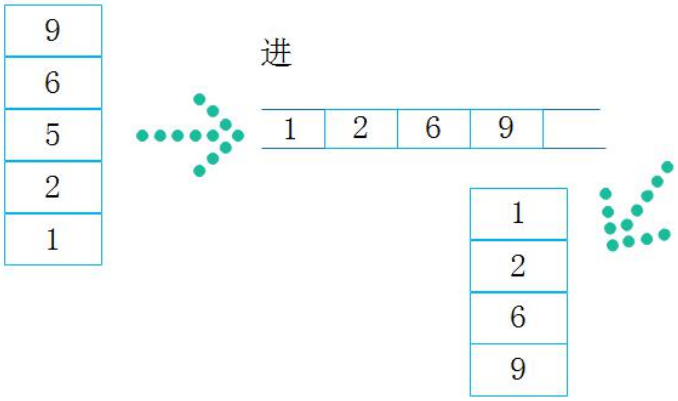
(25,23,34,48,50)(54)(64)(90,89)

(23),(25),(34),(48),(50,)(54),(64),(89),(90)

15、

```
DelElem(LinkList list){
pNode p,q,r;
For(p=list->Link;q!=NULL;q=q->Link){
r->Link;q->Mink;
free(q);
q=r;
}else{
r=r->Link;
}
}
```

16、（1）结果是：1269“1”为栈顶,过程如下图：



demo 的功能：实现删除 e（5）元素，并实现栈 S 数据逆置

17、

（1）

散列地址	0	1	2	3	4	5	6
关键字	63	36	15	22		40	
比较次数	1	1	2	3		1	

（2）ASL=（1+1+2+3+1）/5=1.6

18、m=3--->value=un(3-1)+8=un(2)+8

un(2)=un(2-1)+8=un(1)+8

un(1)=un(0)+8

其中 un(0)=3

un(1)=un(0)+8=3+8=11

un(2)=un(2-1)+8=un(1)+8=11+8=19

value=un(3-1)+8=un(2)+8=19+8=27

最后 return value;value 的值为 27

19.DFS:13452

BFS:13245