堆排序中使用的堆逻辑结构上属于( )

已保存

A.顺序表

B.完全二叉树

C.满二叉树

D.链表

对一个单链表，不借助外部指针的情况下，在表尾插入一个新结点的时间代价为( )

已保存

A.Θ(1)

B.Θ(n)

C.Θ(n2)

D.Θ(nlogn)

一个栈的入栈序列是a,b,c,d,e，则不可能的输出序列是( )

已保存

A.e,d,c,b,a

B.d,e,c,b,a

C.c,d,e,b,a

D.a,b,e,c,d

外部查找算法的主要目标应为( )

已保存

A.减少比较次数

B.减少数据移动次数

C.减少存储空间占用

D.减少磁盘访问次数

一般情况下，当线性表的元素数目( )，最好使用链表实现

已保存

A.较大

B.变化较大

C.较小

D.变化较小

从不带头结点，栈顶指针为top的链栈中进行出栈操作，并用x保存被删除结点的值，则执行( )

已保存

A.`x = top->next->val; top = top->next`

B.`x = top->val; top->next=top->next->next`

C.`top = top->next; x = top->val`

D.`x = top->val; top = top->next`

对任何一棵二叉树T，如果其叶结点的个数为400，度为2的结点个数为( )

已保存

A.200

B.396

C.401

D.399

对结点数量一定的二叉树，高度最小的应为( )

已保存

A.平衡二叉树

B.完全二叉树

C.二叉查找树

D.哈夫曼树

设树的度为4，其中度为1、2、3、4的结点数分别为40、30、20、10个，则该树共有( )个叶子结点

已保存

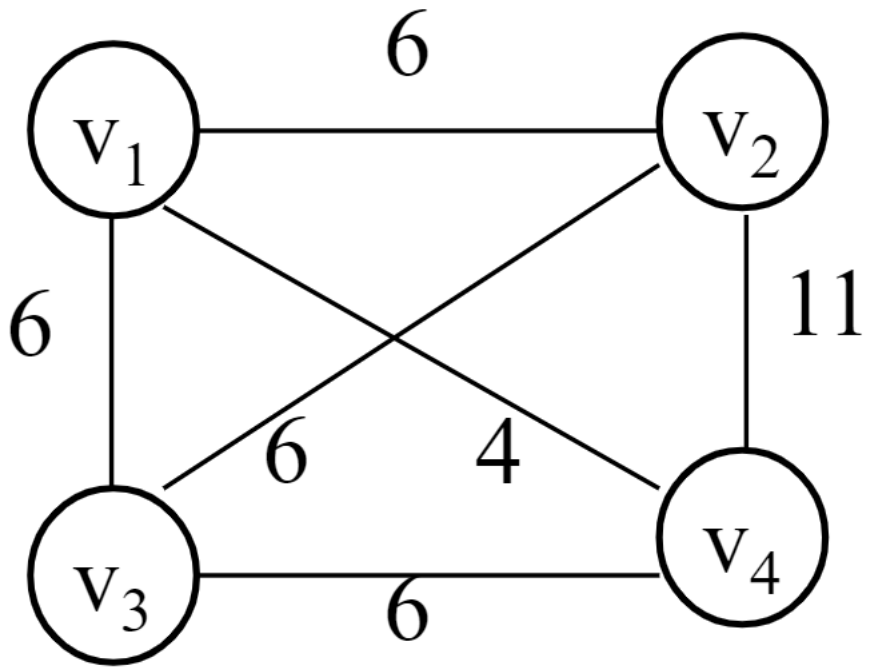
A.200

B.100

C.101

D.无法确定

求下面题图的最小生成树时，可能是克鲁斯卡(Kruskal)算法第二次选中但不是普里姆(Prim)算法(从v_4 开始)第2 次选中的边是( )



已保存

A.(v_1, v_3)

B.(v_1, v_2)

C.(v_2, v_3)

D.(v_3, v_4)

由3个结点所构成的二叉树有_____种形态

已保存 辅助编辑器 预览

(1)

算法的有穷性是指_____.

保存 辅助编辑器 预览

(1)

对于n(n ≥ 0)个元素构成的线性表L，当需要频繁进行随机查找操作时，适合采用存储结构是_____

已保存 辅助编辑器 预览

(1)

有一个循环队列存放在一维数组M[45]中，队列的头指针（指向最早未出队元素）是front，尾指针（指向最新入队的元素的下一个位置）是rear，如果队列中元素的个数为35，front的值是28，则rear的值是_____

已保存 辅助编辑器 预览

(1)

表达式a+b*(c+d*e)的波兰式是

已保存 辅助编辑器 预览

(1)

已知串S="abaabaab"，假设串位置（数组下标）从1开始，其next值为_____

已保存 辅助编辑器 预览

(1)

给定一个关键字序列{2,3,8,9,5,4,1}，以第一个数为枢轴，快速排序第一趟的结果为_____.

已保存 辅助编辑器 预览

(1)

采用折半查找法，在有序表{14，19，31，45，56，79，86，99}中查找元素14和79时，其查找长度分别为___(1)___和___(2)___

已保存 辅助编辑器 预览

(1)
查找14时：

(2)
查找79时：

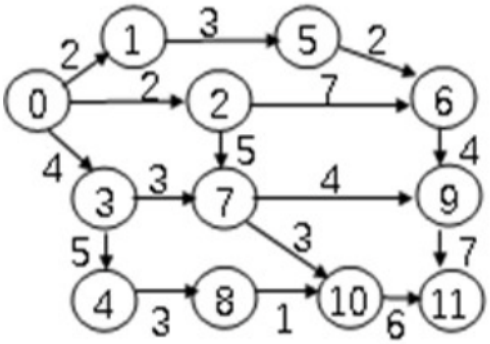
在各种查找方法中，平均查找长度与结点个数n无关的查找方法是

已保存 辅助编辑器 预览

若要求一个稠密图G的最小生成树，最好用_____算法来求解

已保存 辅助编辑器 预览

某一工程作业的网络图如图所示，其中箭头表示作业，箭头边的数字表示完成作业所需的天数。箭头前后的圆圈表示事件，圆圈中的数字表示事件的编号。用事件编号的序列（例如0-2-7-9-11）表示进行作业的路径。请回答下述问题



已保存　辅助编辑器　预览

(1)

用邻接矩阵表示该图，绘制该矩阵. 结点之间不存在边填空格：

	0	1	2	3	4	5	6	7	8	9	10	11
0												
1												
2												
3												
4												
5												
6												
7												
8												
9												
10												
11												

(2)
计算从0点到其他每个顶点的最短路径的长度：

结点	1	2	3	4	5	6	7	8	9	10	11
距离											

(3)
各事件（顶点）的Ve（Vj）和Vl (Vj)的函数值：

结点	0	1	2	3	4	5	6	7	8	9	10	11
Ve												
Vl												

(4)
写出从0点到11的关键路径并计算关键路径的长度

对序列{f, c, i, e, h, a, g, d, j, b}进行排序，请写出

已保存 辅助编辑器 预览

(1)
希尔排序第一趟排序的结果，gap=5

(2)
选择排序第一趟排序的结果

(3)
降序排序的堆排序，第一次建**小顶堆**的结果，用顺序存储表示

Description

程序设计在线评测系统有一个后台评测机，通常是以任务队列的形式处理评测任务。但有时如果管理员未能正确配置，则可能评测机会按**栈**的形式处理任务.

某天有一场比赛管理员把评测机工作模式设为**栈**模式了，给出一系列评测指令，按栈模式的评测顺序输出执行评测任务的id.

Input

数据第一行一个整数 n， 表示有 n 个评测指令，接下来 n 行每行一个评测指令， 1 ≤ n ≤ 1000.

评测指令有如下三种：

- add：新增一个评测任务，第一个评测任务 id 为 1，每次add的评测任务的id都在上一个新增基础上加1；
- rejudge x：对id为x的评测任务重测，数据保证 rejudge 的x是已经出现过的任务id；
- judge：进行一轮评测，按评测机工作模式对上一次judge之后的各任务id进行评测，如果是第一次judge，则从开始的任务算起. 如果没有 judge 指令，则不会执行任何评测.

Output

按**栈**工作模式顺序输出全部指令下执行评测的任务id，每行一个.

Sample

#0

Input Copy

```
5
add
add
add
rejudge 3
judge
```

Output Copy

```
3
3
2
1
```

#1

Input Copy

```
8
add
judge
add
add
add
add
add
judge
```

Output Copy

```
1
6
5
4
3
2
```

Hint

对于样例#0,前三个add添加了任务1、2、3，rejudge增加了一个任务3，按栈评测的顺序先评测rejudge的3，总体顺序是3,3,2,1.

对于样例#1,首先对第一个add的任务1进行了评测，然后对任务2~6进行评测，按栈模式是6,5,4,3,2.

Description

给出一个字符串，按字符出现的顺序构造一棵普通的二叉查找树，并按要求输出对应的内容.

Input

一个仅由小写英文字母组成的字符串，长度不超过100.

忽略重复出现的字母.

Output

基于构造的二叉查找树：

- 输出这个二叉查找树的层次遍历的结果；
- 对于字符串中存在的每个字母，输出根结点到这个字母所在结点路径的字母，按字母的ASCII顺序每个一行输出，格式参考样例.

Sample

#0

Input Copy

```
eacfc
```

Output Copy

```
eafc
a:ea
c:eac
e:e
f:ef
```

#1

Input Copy

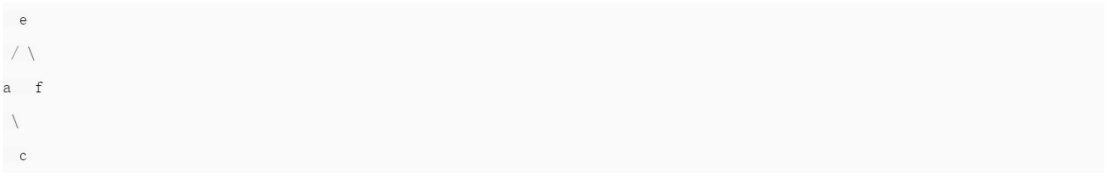
```
ccddf
```

Output Copy

```
cdf
c:c
d:cd
f:cdf
```

Hint

对于样例 #0，构造的二叉查找树为：



其层次遍历为`eafc`，按其ASCII顺序`a, c, e, f`输出4行.

Description

以无向边的形式给出一棵树及其边权，树上两结点之间的唯一路径异或值定义为路径上所有边权按位异或的结果.

求这些路径异或值中的最大值.

Input

第一行给出结点个数 n，$1 \le n \le 10000$.

接下来n − 1行，每行为空格隔开的三个整数u, v, w 表示结点u与结点v之间有一条权值为w的边，$1 \le u, v \le n$，$1 \le w < 2^{31}$.

Output

路径异或值中的最大值.

Sample

#0

Input Copy

```
5
1 2 1
2 5 1
5 3 7
3 4 10
```

Output Copy

```
13
```

#1

Input Copy

```
6
1 4 8
4 3 1
3 5 6
1 6 1
4 2 3
```

Output Copy

```
15
```