1. [单选题]

若图G采用邻接表存储，邻接表中有奇数个表结点，下列选项正确的是\_\_\_\_\_\_\_\_\_\_。

G中有奇数个顶点

G中有偶数个顶点

G为无向图

G为有向图

2. [单选题]

SVM模型通过最大化边界实现线性分类， 以下哪个可以使得SVM实现非线性分类

kernel

松弛变量

对偶空间求解

SMO算法

3. [单选题]

以下关于过拟合和欠拟合说法正确的是

过拟合一般表现为偏差较大

欠拟合一般表现为方差较大

过拟合可以通过减少变量来缓解

欠拟合可以通过正则化来解决

4. [单选题]

以下算法属于有监督算法的是

LDA（latent dirichlet allocation）

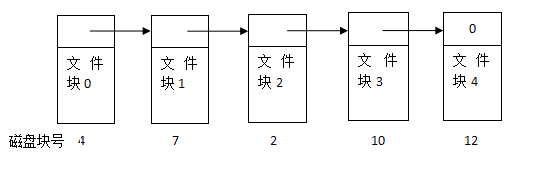
PLSA（probabilistic latent semantic analysis）

knn

SOM(Self-Organizing Maps)

5. [单选题]

某文件系统采用链接存储方式，文件A在磁盘中存放的情况如图所示。



若该文件所在的目录文件已经在内存中，要读取文件块2，需要访问磁盘的次数为

1次

2次

3次

4次

6. [单选题]

以下哪种排序算法在最坏情况下的时间复杂度最小？

冒泡排序

选择排序

归并排序

插入排序

7. [单选题]

两台主机A和B已建立了TCP连接，A始终以MSS=1KB大小的段发送数据，并一直有数据发送；B每收到一个数据段都会发出一个接收窗口为9KB的确认段。

若A在T时刻发生超时时拥塞窗口为8KB，则从T时刻起，不再发生超时的情况下，经过10个RTT后，A的发送窗口是 （）

8KB

9KB

10KB

11KB

8. [单选题]

设栈S初始状态为空。元素1,2,3,4,5,6依次通过栈S，若出栈的顺序为4,6,5,3,2,1，则栈S的容量至少应该为？

3

4

5

6

9. [单选题]

在Linux系统中，因为某些原因造成了一些进程变成孤儿进程，那么这些孤儿进程会被以下哪一个系统进程接管？

sshd

init

top

syslogd

10. [单选题]

在软件开发中，经典的模型就是瀑布模型，下列关于瀑布模型的说法正确的是()

瀑布模型具由于良好的灵活性

瀑布模型采用结构化的分析与设计方法，将逻辑实现与物理实现分开

瀑布模型的核心是按照软件开发的时间顺序将问题简化

利用瀑布模型，如果发现问题则修改的代价很低

11. [编程题] 缺失的括号

时间限制：1秒

空间限制：65536K

一个完整的括号字符串定义规则如下:

1、空字符串是完整的。

2、如果s是完整的字符串，那么(s)也是完整的。

3、如果s和t是完整的字符串，将它们连接起来形成的st也是完整的。

例如，"(()())", ""和"(())()"是完整的括号字符串，"())(", "()(" 和 ")"是不完整的括号字符串。

牛牛有一个括号字符串s,现在需要在其中任意位置尽量少地添加括号,将其转化为一个完整的括号字符串。请问牛牛至少需要添加多少个括号。

输入描述:

输入包括一行,一个括号序列s,序列长度length(1 ≤ length ≤ 50).

s中每个字符都是左括号或者右括号,即'('或者')'.

输出描述:

输出一个整数,表示最少需要添加的括号数

输入例子1:

(()(()

输出例子1:

2

12. [编程题] 最后一位

时间限制：1秒

空间限制：65536K

牛牛选择了一个正整数X,然后把它写在黑板上。然后每一天他会擦掉当前数字的最后一位,直到他擦掉所有数位。 在整个过程中,牛牛会把所有在黑板上出现过的数字记录下来,然后求出他们的总和sum.

例如X = 509, 在黑板上出现过的数字依次是509, 50, 5, 他们的和就是564.

牛牛现在给出一个sum,牛牛想让你求出一个正整数X经过上述过程的结果是sum.

输入描述:

输入包括正整数sum(1 ≤ sum ≤ 10^18)

输出描述:

输出一个正整数,即满足条件的X,如果没有这样的X,输出-1。

输入例子1:

564

输出例子1:

509

13. [编程题] 冒泡排序

时间限制：1秒

空间限制：65536K

牛牛学习了冒泡排序,并写下以下冒泡排序的伪代码,注意牛牛排序的数组a是从下标0开始的。

BubbleSort(a):

Repeat length(a)-1 times:

For every i from 0 to length(a) - 2:

If a[i] > a[i+1] then:

Swap a[i] and a[i+1]

牛牛现在要使用上述算法对一个数组A排序。

在排序前牛牛允许执行最多k次特定操作(可以不使用完),每次特定操作选择一个连续子数组,然后对其进行翻转,并且k次特定操作选择的子数组不相交。

例如A = {1, 2, 3, 4, 5, 6, 7}, k = 1,如果牛牛选择的子数组是[2,4](注意下标从0开始),那么翻转之后的数组变为A = {1, 2, 5, 4, 3, 6, 7}。

牛牛知道冒泡排序的效率一定程度上取决于Swap操作次数,牛牛想知道对于一个数组A在进行k次特定操作之后,再进行上述冒泡排序最少的Swap操作次数是多少?

输入描述:

输入包括两行,第一行包括两个正整数n和k(2 ≤ n ≤ 50, 1 ≤ k ≤ 50),表示数组的长度和允许最多的特定操作次数。

第二行n个正整数A[i](1 ≤ A[i] ≤ 1000),表示数组内的元素,以空格分割。

输出描述:

输出一个整数,表示在执行最多k次特定操作之后,对数组进行上述冒泡排序需要的Swap操作次数。

输入例子1:

3 2

2 3 1

输出例子1:

1

答案：

DACCC

CBCBB

11题：

#include

#include

using namespace std;

int main()

{

    string str; cin >> str;

    int result = 0, num = 0;

    for (int i = 0; i < str.length(); i++) {

        if (str[i] == '(') num++;

        else {

            if (num == 0) result++;

            else num--;

        }

    }

    cout << result + num << endl;

    return 0;

}

12. #include<stdio.h>

long long getSum(long long);

int main(){

long long sum,l,r,mid;

//freopen("input.txt","r",stdin);

scanf("%lld",&sum);

for(l=0,r=sum;l+1<r;){

mid=l+(r-l)/2;

if(getSum(mid)==sum){

printf("%lld",mid);

return 0;

}else if(getSum(mid)<sum) l=mid;

else r=mid;

}

if(getSum(l)==sum) printf("%lld",l);

else if(getSum(r)==sum) printf("%lld",r);

else printf("-1");

}

long long getSum(long long x){

long long sum=0;

while(x!=0) sum+=x,x/=(long long)10;

return sum;

}//数据这么大，一看就是二分啦~

13. #include <stdio.h>

#include <vector>

#include <algorithm>

using namespace std;

const int maxn=55;

int getCnt(vector<int> A,int x){

int n=A.size(),cnt=0;

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

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

if(A[j]>A[i]) cnt++;

return cnt;

}

int n,dp[maxn][maxn],k,K,i,j;

int main(){

scanf("%d%d",&n,&K);

vector<int> A(n);

for(i=0;i<n;i++) scanf("%d",&A[i]);

for(i=n-1;i>=0;i--)

for(k=0;k<=K;k++){

vector<int> tmp1(A.begin(),A.begin()+i+1);

dp[i][k]=getCnt(tmp1,i)+dp[i+1][k];

if(k>=1){

for(j=i+1;j<n;j++){

vector<int> tmp2(A.begin(),A.begin()+j+1);

reverse(tmp2.begin()+i,tmp2.begin()+j+1);

dp[i][k]=min(dp[i][k],getCnt(tmp2,i)+dp[j+1][k-1]);

}

}

}

printf("%d",dp[0][K]);

}