**算法训练 单词接龙**

时间限制：1.0s   内存限制：256.0MB

**问题描述**  
  
　　单词接龙是一个与我们经常玩的成语接龙相类似的游戏，现在我们已知一组单词，且给定一个开头的字母，要求出以这个字母开头的最长的“龙”（每个单词都最多在“龙”中出现两次），在两个单词相连时，其重合部分合为一部分，例如 beast和astonish，如果接成一条龙则变为beastonish，另外相邻的两部分不能存在包含关系，例如at 和 atide 间不能相连。  
  
**输入格式**  
  
　　输入的第一行为一个单独的整数n (n<=20)表示单词数，以下n 行每行有一个单词，输入的最后一行为一个单个字符，表示“龙”开头的字母。你可以假定以此字母开头的“龙”一定存在.  
  
**输出格式**  
  
　　只需输出以此字母开头的最长的“龙”的长度  
  
**样例输入**  
　　5  
　　at  
　　touch  
　　cheat  
　　choose  
　　tact  
　　a

样例输出

23

样例说明

　　连成的“龙”为atoucheatactactouchoose

锦囊1

搜索。

锦囊2

使用递归搜索，每次试着将一个可以加到后面的单词加到后面，并检查答案是不是更优。

本题的C++参考代码如下：

#include <cstdio>

#include <cmath>

#include <cstring>

#include <iostream>

#include <algorithm>

using namespace std;

using namespace std ;

int map[25][25];

int n,cnt;

char c;

char st[25][25];

int visit[25];

int dfs(int s)

{

int temp,maxlen=0,t=0;

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

{

temp=0;

if(map[s][i]&&visit[i])

{

t=1;

visit[i]--;

temp=dfs(i);

visit[i]++;

temp=temp+strlen(st[s])-map[s][i];

}

if(temp>maxlen) maxlen=temp;

}

if(!t) return strlen(st[s]);

return maxlen;

}

int main()

{

while(scanf("%d",&n)!=EOF)

{

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

scanf("%s",st[i]);

getchar();

scanf("%c",&c);

memset(map,0,sizeof(map));

int len1,len2;

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

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

{

len1=strlen(st[i]);

len2=strlen(st[j]);

for(int k=0;k<len1&&k<len2;k++)

{

if(strncmp(st[i]+len1-k-1,st[j],k+1)==0)

{

map[i][j]=k+1;

break;

}

}

}

for(int i=0;i<n;i++) visit[i]=2;

cnt=0;

int temp;

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

{

if(st[i][0]==c)

{

visit[i]--;

temp=dfs(i);

if(temp>cnt) cnt=temp;

visit[i]++;

}

}

printf("%d\n",cnt);

}

}

本题的C参考代码如下：

#include <stdio.h>

#include <string.h>

int Len[21],P[21],MAX,N;

char Word[21][100];

void DFS(int d,int s)

{

int i,j,k,l;

for (i=1;i<=N;i++)

if (P[i]<2)

for (j=0;j<Len[d];j++)

if (Word[i][0]==Word[d][j])

{

for (l=1,k=j+1;k<Len[d]&&Word[i][l]==Word[d][k];k++,l++);

if (k<Len[d]) continue;

P[i]++;

DFS(i,s+Len[i]-l);

P[i]--;

}

if (s>MAX) MAX=s;

}

int main()

{

int i;

scanf("%d",&N);

for (i=1;i<=N;Len[i]=strlen(Word[i]),i++) scanf("%s",Word[i]);

scanf("%s",Word[0]);

Len[0]=strlen(Word[0]);

DFS(0,Len[0]);

printf("%d",MAX);

return 0;

}

本题的Java参考代码如下：

import java.util.Scanner;

public class Main {

private static String[] a = new String[20];

private static int[] b = new int[20];

private static int max;

private static int n;

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

n = scanner.nextInt();

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

a[i] = scanner.next();

String string = scanner.next();

f(string, string.length());

System.out.println(max);

}

private static void f(String s, int length) {

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

if (a[i].indexOf(s) == 0 && b[i] < 2) {

int length1 = s.length();

int length2 = a[i].length();

b[i]++;

int p = 1;

length = length + length2 - length1;

while (p < length2) {

f(a[i].substring(length2 - p, length2), length);

p += 1;

}

length = length - length2 + length1;

b[i]--;

}

max = length > max ? length : max;

}

}