

2020/07/18--2020/07/24

团队训练

暂无

王瑞琦

比赛

无

专题

无

冯宇扬

没有

常程

比赛

没有

专题

这一周在刷AC自动机的题来着，所以没有新的东西

本周推荐

王瑞琦

翻出来以前写的一道模拟退火入门题，温习了一下。

来源

洛谷P1337 [平衡点](#)

标签

模拟退火，数学

题意

给出桌面上 n 个洞的坐标 n 个重物系绳分别穿过 n 个洞，求绳结最后的平衡位置。

题解

绳结平衡，即所受 n 个力达到平衡。

而当绳结处于一个非平衡点时，其所受合力方向必指向最后的平衡点，因此我们将绳结向该方向移动一段距离。

循环上述过程，每次移动的距离逐渐减少，最终绳结将“逼近”最后的平衡点。

```
#include <stdio.h>
#include <math.h>
#define abs(x) ((x) >= 0 ? (x) : (-(x)))
#define true 1
#define false 0

int main()
{
    int a[34][1001]={0};
    int n;
    int XF=true, YF=true;
    scanf("%d",&n);
    for (int i=1;i<=n;i++)
        scanf("%d%d%d",&a[1][i],&a[2][i],&a[3][i]);
    double t=5000,tx,ty;
    double x=1,y=1;
    while (true)
    {
        tx=x;
        ty=y;
        double lx=0,ly=0,dist;
        for (int i=1;i<=n;i++)
        {
```

```
dist=sqrt((a[2][i]-y)*(a[2][i]-y)+(a[1][i]-x)*(a[1][i]-x));
if (dist==0) continue;
lx=lx+a[3][i]/dist*(a[1][i]-x);
ly=ly+a[3][i]/dist*(a[2][i]-y);

}
dist=sqrt(lx*lx+ly*ly);
x=x+t/dist*lx;
y=y+t/dist*ly;
if (abs(tx-x)<0.00001&&abs( ty - y)<0.00001) break;
if ((XF!=(x>tx))||(YF!=(y>ty))){
    XF=1-(x>tx);
    YF=1-(y>ty);
    t=t*0.9;
}
}
printf("%.3lf %.3lf",x,y);
getchar();
getchar();
return 0;
}
```

comment

比较入门的一道题，复习的时候却调了好久=。=
关于模拟退火还在学习当中...

冯宇扬

本周完全摸鱼

常程

来源

[p4052 文本生成器](#)

标签

AC自动机 DP

题意

给出一些模式串和一个长度，求出给定长度下至少包含一个给出的模式串的文本串数量。

思路

这道题可以通过计算不包含任何一个模式串的文本串数量间接求解。
主要是在Trie树上运用dp

在这里我想做题的人需要对文本串在Trie树上匹配的过程有一个直观而清晰的认识——匹配到当前的树上的位置代表一切其fail一串上的均可以匹配，但以该点为fail的位置不能被匹配。

运用dp $\square$ dp[i][j]表示 长度为i结点序号为j的位置不包含任何一个模式串的文本串数量。

状态转移方程为 $\square$ dp[i+1][son[j][k]]+=dp[i][j] (!flag[son[j][k]])

这里面需要注意的是，如果一个结点是一个模式串的末尾，那么以其为fail关系的子孙的结点都要标记flag $\square$ 原因如上。

```
#include<cstdio>
#include<queue>
using namespace std;
int n,m,tot=1,dp[6005][105],mod=10007;
char s[105];
struct unit{int s[27],fail,flag;}T[6005];
void Insert()
{
    int root=1,id;
    for(int i=0;s[i];i++)
    {
        id=s[i]-65;
        if(!T[root].s[id])
            T[root].s[id]=++tot;
        root=T[root].s[id];
    }
    T[root].flag=1;
}
void get_fail()
{
    for(int i=0;i<26;i++)
        T[0].s[i]=1;
    T[1].fail=0;
    queue<int>q;
    q.push(1);
    int m=1,head;
    while(m--)
    {
        head=q.front();
        q.pop();
        for(int i=0,v,fafail;i<26;i++)
        {
            v=T[head].s[i];
            fafail=T[head].fail;
            if(!v) T[head].s[i]=T[fafail].s[i];
            else
```

```
        {
            T[v].fail=T[fafail].s[i];
            T[v].flag|=T[T[v].fail].flag;
            q.push(v);
            m++;
        }
    }
}

int main()
{
    scanf("%d%d",&n,&m);
    for(int i=1;i<=n;i++)
    {
        scanf("%s",s);
        Insert();
    }
    get_fail();
    dp[1][0]=1;
    for(int i=1;i<=m;i++)
        for(int j=1;j<=tot;j++)
            for(int k=0;k<26;k++)
                if(!T[T[j].s[k]].flag)
                    dp[T[j].s[k]][i]=(dp[T[j].s[k]][i]+dp[j][i-1])%mod;

    int ans=1;
    for(int i=26,j=m;j>=1,i=(i*i)%mod)
        if(j&1)
            ans=(ans*i)%mod;
    for(int i=1;i<=tot;i++)
        ans=(ans-dp[i][m]+mod)%mod;
    printf("%d",ans);
    return 0;
}
```

comment

还算是有点复杂的，当时调了一段时间。

From:

<https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link:

https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:no_morning_training:weekly:week8

Last update: 2020/07/29 09:01

