

[解题报告] POJ1226

[Source] <http://poj.org/problem?id=1226>

[Description]

给出 $n(n \leq 100)$ 个长度不超过 100 的字符串，求这些串的最长公共子串（该子串本身或者它翻转后的串在所有给出的串中出现）。

[Solution]

暴力枚举。首先找出长度最小的串，然后枚举它的所有子串，判断该子串或其倒转后是否是所有串的子串。理论上时间复杂度为 $l^3 * n = 100^4 = 10^8$ ，实际测试数据会远小于此，好在没有构造极限数据来卡时间。

[Code]

```
#include<cstdio>
#include<cstring>
using namespace std;

char str[220][110];
int len[220];

int check(int p, int s, int l, int t)
{
    int i, j, flag;

    for (i=0; i+l-1<len[t]; i++)
    {
        flag=1;
        for (j=0; j<l; j++)
            if (str[p][s+j] != str[t][i+j])
            {
                flag=0;
                break;
            }

        if (flag)
            return 1;
    }
}
```

```

    }

    return 0;
}

int main()
{
    int t,n,min,i,j,l,ans,flag;

    scanf("%d",&t);
    while (t--)
    {
        scanf("%d",&n);
        for (i=0;i<n;i++)
        {
            scanf("%s",&str[i]);
            l=strlen(str[i]);
            for (j=0;j<l;j++)
                str[i+n][j]=str[i][l-j-1];
            str[i+n][l]=0;

            len[i]=len[i+n]=l;
        }

        min=0;
        for (i=1;i<n;i++)
            if (len[min]>len[i])
                min=i;
        ans=0;
        for (l=len[min];l>0;l--)
        {
            for (i=0;i+l-1<len[min];i++)
            {
                flag=1;
                for (j=0;j<n;j++)
                {
                    if (j==min)
                        continue;

                    if
(check(min,i,l,j)==0&&check(min,i,l,j+n)==0)
                {

```

```
        flag=0;
        break;
    }
}

if (flag)
{
    ans=1;
    break;
}

if (ans)
    break;
}

printf("%d\n",ans);
}

return 0;
}
```