

分数:100+5+0=105

A:

题意:

求 a, b 使 $a+b=n$ 且 $\text{lcm}(a, b)$ 最小。

思路:

设 $a < b$

不难得出, $\text{lcm}(a, b)$ 小于 n , 且 a 肯定为 n 的约数

Sqrt(n)暴力搜索 n 的约数就行了

时间复杂度: $O(\sqrt{n})$

代码:

```
#include <bits/stdc++.h>

using namespace std;

typedef long long ll;

int n, ans, sum = 0;

int gcd(int x, int y){
    return y == 0 ? x : gcd(y, x % y);
}

int main(){
    scanf("%d", &n);

    if(n % 2 == 0) printf("%d %d\n", n / 2, n / 2);

    else{
        for(int i = 1; i <= sqrt(n); i++)
```

```

        if(n%i==0){

            int p1=i,p2=n/i,gcd1=gcd(p1,n-p1),gcd2=gcd(p2,n-p2);

            if(gcd1>sum){

                sum=gcd1;ans=p1;

            }

            if(p2!=n && gcd2>sum && p2<=n/2){

                sum=gcd2;ans=p2;

            }

        }

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

    }

    return 0;

}

```

B:

题意:

给两个字符串 s、t，每一次操作可以把 s 的一个字符插进它之前的位置里，求最少需要操作次数，使 s 变成 t（不可行输出 -1）

思路:

先统计两个字符串的小写字母的个数

然后我们求一遍 LCS

但是和普通的 LCS 不同，由于加了条件，我们必须进行特殊处理

当 s 里的 a 字符的个数小于 t 里 a 字符的个数时，才能有效转移 $dp[i][j-1]$

时间复杂度 $O(n^2)$

代码:

```
#include<bits/stdc++.h>

using namespace std;

const int N=2005;

const int inf=1e9+7;

char s[N],t[N];

int T,n,cnt[2][N][30],dp[N][N];

int main(){

    scanf("%d", &T);

    while(T--){

        scanf("%d", &n);

        scanf("%s", s+1);

        scanf("%s", t+1);

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

            for(int j = 1; j <= 26; j++) cnt[0][i][j]=cnt[1][i][j]=0;

        for(int i = 1; i <= n; i++)

            for(int j = i; j <= n; j++) dp[i][j]=inf;

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

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

            for(int j = 1; j <= 26; j++){

                cnt[0][i][j]+=cnt[0][i-1][j];
```

```

        cnt[1][i][j]+=cnt[1][i-1][j];

    }

    cnt[0][i][s[i]-'a'+1]++;

    cnt[1][i][t[i]-'a'+1]++;

}

bool flag=1;

for(int i = 1; i <= 26; i++)

    if(cnt[0][n][i]!=cnt[1][n][i]){

        flag=0;

        puts("-1");

        break;

    }

if(!flag) continue;

for(int i = 1; i <= n; i++)

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

        dp[i][j]=dp[i-1][j]+1;

        if(s[i]==t[j]) dp[i][j]=min(dp[i][j],dp[i-1][j-1]);

        if(cnt[0][i][t[j]-'a'+1]<cnt[1][i][t[j]-'a'+1])

dp[i][j]=min(dp[i][j],dp[i][j-1]);

    }

if(dp[n][n]<inf)printf("%d\n", dp[n][n]);

else printf("%d\n", -1);

```

```

    }

    return 0;

}

```

错误：

没有想出正解，且暴力打挂了。

C：

题意：

给两个括号序列 s, t ，求一个合法的括号序列，使得 s 和 t 是它的子字符串，且长度最短

思路：

设 $dp[i][j][k]$ 表示在 s 的 i 位置和 t 的 j 位置，左括号比右括号多了 k 个的情况数

设 $li=i+(i \leq l1 \ \&\& \ s1[i]='('), lj=j+(j \leq l2 \ \&\& \ s2[j]='('); ri=i+(i \leq l1 \ \&\& \ s1[i]==')'), rj=j+(j \leq l2 \ \&\& \ s2[j]==')');$

$dp[i][j][k] = \min\{dp[li][lj][k+1] (k \leq n+m), dp[ri][rj][k-1] (k > 0)\}$

(我这里用 dfs 求的)

最后用数组储存字符，输出答案

时间复杂度 $O(n^3)$

代码：

```

#include<bits/stdc++.h>

using namespace std;

const int INF=1e9+7;

struct node{

```

```

    int i,j,k;

}nxt[205][205][205];

char s1[205],s2[205],ch[205][205][205];

int dp[205][205][205],l1,l2;

int dfs(int i, int j, int k){

    if(dp[i][j][k]) return dp[i][j][k];

    if(i==l1+1 && j==l2+1) return k;

    int li=i+(i<=l1 && s1[i]=='('),lj=j+(j<=l2 && s2[j]=='(');

    int ri=i+(i<=l1 && s1[i]==')'),rj=j+(j<=l2 && s2[j]==')');

    int ret1=INF,ret2=INF;

    if(k<=max(l1,l2)) ret1=dfs(li,lj,k+1)+1;

    if(k>0) ret2=dfs(ri,rj,k-1)+1;

    if(ret1<ret2){

        dp[i][j][k]=ret1;nxt[i][j][k]=(node){li,lj,k+1};ch[i][j][k]='(';

    }else{

        dp[i][j][k]=ret2;nxt[i][j][k]=(node){ri,rj,k-1};ch[i][j][k]==')';

    }

    if(ret1==ret2 && ret1==INF) return INF;

    return dp[i][j][k];

}

void print(int i, int j, int k){

    if(ch[i][j][k]=='(' || ch[i][j][k]==')') printf("%c", ch[i][j][k]);

```

```
if(nxt[i][j][k].i) print(nxt[i][j][k].i,nxt[i][j][k].j,nxt[i][j][k].k);  
  
else for(int i = 1; i <= k; i++) printf(" ");  
  
}
```

```
int main(){  
  
    scanf("%s", s1+1);  
  
    scanf("%s", s2+1);  
  
    l1=strlen(s1+1),l2=strlen(s2+1);  
  
    dfs(1,1,0);  
  
    print(1,1,0);  
  
    puts("");  
  
    return 0;
```

}错误:

第二题花了太长时间，来不及写了