T1 比较大小 解题报告

【解题思路】

把串全部变为小写，按题意模拟即可。

【参考代码】

```
#include<bits/stdc++.h>
using namespace std;

int main()
{
    freopen("lexi.in","r",stdin);
    freopen("lexi.out","w",stdout);
    string a, b;
    cin >> a >> b;
    int len = a.length();
    for(int i = 0; i < len; i++)
    {
        if(a[i] >= 'A' && a[i] <= 'Z')  a[i] = a[i] - 'A' + 'a';
        if(b[i] >= 'A' && b[i] <= 'Z')  b[i] = b[i] - 'A' + 'a';
        if(a[i] == b[i])  continue ;
        puts((a[i] < b[i]) ? "-1" : "1");
        return 0;
    }
    puts("0");
    return 0;
}
```

T2 染色 解题报告

【解题思路】

首先每个位置竖着涂需要花n次。

然后我们考虑横着涂的情况，显然每一次横着涂一定要与最矮的一条齐平，否则还需要n次竖着涂的，显然更劣。

与最矮的一条齐平横着涂后又会产生一些子问题，递归进行 DP 处理即可。

【参考代码】

```
#include<bits/stdc++.h>
using namespace std;

int n, a[5005], f[5005][5005];

int solve(int l, int r, int h)
{
    int mpos[5005], cnt = 0;
    if(l > r)  return f[l][r] = 0;
    if(l == r)  return f[l][r] = 1;
    int minn = 1E9;
    for(int i = l; i <= r; i++)
        minn = min(minn, a[i]);
    int sum = minn - h;
    for(int i = l; i <= r; i++)  if(a[i] == minn)  mpos[++cnt] = i;
    solve(l, mpos[1] - 1, minn), sum += f[l][mpos[1] - 1];
    for(int i = 1; i < cnt; i++) {
        solve(mpos[i] + 1, mpos[i + 1] - 1, minn);
        sum += f[mpos[i] + 1][mpos[i + 1] - 1];
    }
    solve(mpos[cnt] + 1, r, minn), sum += f[mpos[cnt] + 1][r];
    return f[l][r] = min(sum, r - l + 1);
}

int main()
{
    freopen("paint.in","r",stdin);
    freopen("paint.out","w",stdout);
    scanf("%d", &n);
    for(int i = 1; i <= n; i++)  scanf("%d", &a[i]);
    solve(1, n, 0);
    cout << f[1][n] << endl;
    return 0;
}
```

T3 质因数分解 解题报告

【解题思路】

看到$n \leq 16$，直接想到搜索出所有符合条件的数，但会超时。

考虑折半搜索算法，搜索出前一半数可以构造的符合条件的数与后一半数可以构造出的符合条件的数，然后两类数两两相乘即可得出所有符合条件的数。

但是一个一个乘起来找第k个还是太慢。我们考虑二分一个数x，如果两类数两两相乘的值有多于k个小于等于x，那我们就认为其符合条件，然后缩小范围直到找出最终答案。

考虑如何找出小于等于x的乘积的个数，只需要将两类数分别排序，然后使用双指针算法找出对于每一个第一类数有多少个第二类数符合条件，将所有的数目相加即可。

【参考代码】

```
#include<bits/stdc++.h>
using namespace std;

const long long Mx = 1E18;
int n, k, a[25];
vector<long long> mul[2];

void dfs(int nw, int lim, long long M, int typ)
{
    if(nw == lim + 1)  return mul[typ].push_back(M), void();
    if(Mx / a[nw] >= M)  dfs(nw, lim, M * a[nw], typ);
    dfs(nw + 1, lim, M, typ);
}

int chk(long long x)
{
    int nw = mul[0].size();
    long long ans = 0;
    for(int i = 0; i < mul[1].size(); i++)
    {
        while(nw >= 1 && x / mul[0][nw - 1] < mul[1][i])  --nw;
        ans += nw;
    }
    return ans >= k;
}
```

```
int main()
{
    freopen("prime.in","r",stdin);
    freopen("prime.out","w",stdout);
    cin >> n;
    for(int i = 1; i <= n; i++)  cin >> a[i];
    sort(a + 1, a + n + 1);
    for(int i = 1; i <= n / 2; i++)
        if(i & 1)  swap(a[i], a[i + n / 2]);
    cin >> k;
    dfs(1, n / 2, 1, 0);
    dfs(n / 2 + 1, n, 1, 1);
    sort(mul[0].begin(), mul[0].end());
    sort(mul[1].begin(), mul[1].end());
    long long l = 1, r = 1E18, ans = -1;
    while(l <= r)
    {
        long long mid = (l + r) >> 1;
        if(chk(mid))  ans = mid, r = mid - 1;
        else  l = mid + 1;
    }
    cout << ans << endl;
    return 0;
}
```

T4 滑冰 解题报告

【解题思路】

考虑对于三个点i,j,k，若$x_i < x_j < x_k$，

则$i \to k = x_k - x_i = (x_k - x_j) + (x_j - x_i) = i \to j \to k$，边$i \to k$是根本不用连的。

所以将所有点的x坐标排序，然后从小到大连边即可。

对于y坐标同理。

然后对于限制条件$min|a_x - b_x|, |a_y - b_y|$，只需要跑最短路即可，因为最短路不可能把你导向一条长的路径。

【参考代码】

```
#include<bits/stdc++.h>
using namespace std;

#define MAXN 200005
#define INF 0x3f3f3f3f
long long ans,dis[MAXN];
int head[MAXN],ecnt,n;

struct node{
    int x,y,id;
}a[MAXN];

struct point{
    int y,z,next;
}edge[4*MAXN];

void add(int x,int y,int z)
{
    ecnt++;
    edge[ecnt].y=y;edge[ecnt].z=z;edge[ecnt].next=head[x];
    head[x]=ecnt;
}

int cmp1(const node& a,const node& b)
{
    return a.x<b.x;
}

int cmp2(const node& a,const node& b)
{
    return a.y<b.y;
}
```

```
struct line{
    int x,dis;
    bool operator < (const line &o) const{
        return dis > o.dis;
    }
};

void dijkstra()
{
    priority_queue<line> q;
    for (int i=1;i<=n;i++)
        dis[i]=INF;
    dis[1]=0;
    q.push((line){1,0});
    while (!q.empty())
    {
        line t=q.top();q.pop();
        if (dis[t.x]!=t.dis) continue;
        int p=head[t.x];
        while (p){
            if (dis[edge[p].y]>dis[t.x]+edge[p].z)
            {
                dis[edge[p].y]=dis[t.x]+edge[p].z;
                q.push((line){edge[p].y,dis[edge[p].y]});
            }
            p=edge[p].next;
        }
    }
}
```

```
int main()
{
    freopen("shortest.in","r",stdin);
    freopen("shortest.out","w",stdout);
    scanf("%d", &n);
    for (int i=1;i<=n;i++)
    {
        scanf("%d%d", &a[i].x, &a[i].y);
        a[i].id=i;
    }
    sort(a+1,a+n+1,cmp1);
    for (int i=1;i<n;i++)
    {
        add(a[i].id,a[i+1].id,a[i+1].x-a[i].x);
        add(a[i+1].id,a[i].id,a[i+1].x-a[i].x);
    }
    sort(a+1,a+n+1,cmp2);
    for (int i=1;i<n;i++)
    {
        add(a[i].id,a[i+1].id,a[i+1].y-a[i].y);
        add(a[i+1].id,a[i].id,a[i+1].y-a[i].y);
    }
    dijkstra();
    printf("%d",dis[n]);
}
```