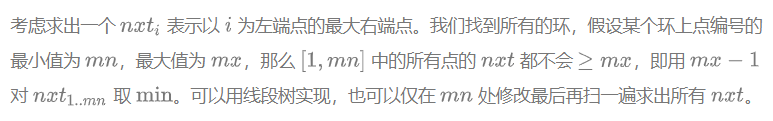
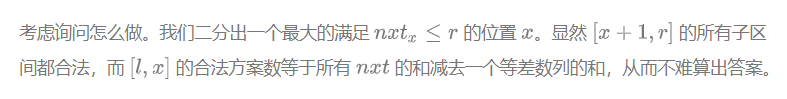
CF901C Bipartite Segments

[Codeforces](http://codeforces.com/problemset/problem/901/C)

[Luogu](https://www.luogu.com.cn/problem/CF901C)

## 分析

「图中没有偶环」等价于「图是仙人掌」。因此本题中「一个诱导子图是二分图」等价于「一个诱导子图没有环」。

## 代码

*// ===================================*

*//   author: M\_sea*

*//   website: http://m-sea-blog.com/*

*// ===================================*

#include <algorithm>

#include <iostream>

#include <cstdlib>

#include <cstring>

#include <cstdio>

#include <vector>

#include <cmath>

#define re **register**

using **namespace** std;

typedef **long** **long** ll;

**inline** **int** read()

{

**int** X = 0, w = 1;

**char** c = getchar();

    while (c < '0' || c > '9')

    {

        if (c == '-')

            w = -1;

        c = getchar();

    }

    while (c >= '0' && c <= '9')

        X = X \* 10 + c - '0', c = getchar();

    return X \* w;

}

**const** **int** N = 300000 + 10, M = 300000 + 10;

**int** n, m, q;

**int** nxt[N];

ll sum[N];

**struct** edge

{

**int** v, nxt;

} e[M << 1];

**int** head[N];

**inline** **void** addEdge(**int** u, **int** v)

{

**static** **int** c = 0;

    e[++c] = (edge){v, head[u]}, head[u] = c;

}

**int** vis[N], in[N], sta[N], top = 0;

**inline** **void** dfs(**int** u, **int** fa)

{

    vis[u] = 1, sta[++top] = u;

    for (re **int** i = head[u]; i; i = e[i].nxt)

    {

**int** v = e[i].v;

        if (v == fa)

            continue;

        if (!vis[v])

            dfs(v, u);

        else if (!in[v])

        {

**int** mx = 0, mn = n + 1;

            while ("longge ak ioi")

            {

**int** x = sta[top--];

                in[x] = 1;

                mx = max(mx, x), mn = min(mn, x);

                if (x == v)

                    break;

            }

            nxt[mn] = min(nxt[mn], mx - 1);

        }

    }

    if (!in[u])

        --top;

}

**int** main()

{

    n = read(), m = read();

    for (re **int** i = 1; i <= m; ++i)

    {

**int** u = read(), v = read();

        addEdge(u, v), addEdge(v, u);

    }

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

        nxt[i] = n;

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

        if (!vis[i])

            dfs(i, 0);

    for (re **int** i = n - 1; i; --i)

        nxt[i] = min(nxt[i + 1], nxt[i]);

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

        sum[i] = sum[i - 1] + nxt[i];

    q = read();

    while (q--)

    {

**int** l = read(), r = read();

**int** L = l, R = r;

        while (L < R)

        {

**int** mid = (L + R + 1) >> 1;

            if (nxt[mid] < r)

                L = mid;

            else

                R = mid - 1;

        }

        if (nxt[L] > r)

            --L;

        printf("%lld\n", 1ll \* (r - L) \* (r - L + 1) / 2 + sum[L] - sum[l - 1] - 1ll \* (l + L - 2) \* (L - l + 1) / 2);

    }

    return 0;

}