#include <cstdio>

#include <algorithm>

using namespace std;

struct P

{

int x, y;

} a[100010], b[100010];

bool operator < (P a, P b)

{

return a.x > b.x || a.x == b.x && a.y > b.y;

}

int d[500010], N;

void update(int p, int v)

{

int l, r;

p += N;

d[p] = v;

p >>= 1;

while (p)

{

l = p<<1;

r = l|1;

if (d[l] > d[r])

d[p] = d[l];

else

d[p] = d[r];

p >>= 1;

}

}

int find(int p, int s, int e, int a, int b, int v)

{

if (d[p] < v)

return -1;

if (s == e)

return p;

int m = (s+e)>>1;

if (b <= m)

return find(p<<1, s, m, a, b, v);

if (a > m)

return find((p<<1)|1, m+1, e, a, b, v);

int r = find((p<<1)|1, m+1, e, m+1, b, v);

if (r != -1)

return r;

return find(p<<1, s, m, a, m, v);

}

int main()

{

freopen("gourmet.in", "r", stdin);

freopen("gourmet.out", "w", stdout);

long long ans=0;

int n, m, p, l, r, i;

scanf("%d%d", &n, &m);

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

scanf("%d%d", &a[i].y, &a[i].x);

for (i = 0; i < m; i++)

scanf("%d%d", &b[i].x, &b[i].y);

sort(a, a+n);

sort(b, b+m);

for (N = 1; N < m; N<<=1);

for (i = 0; i < m; i++)

update(i, b[i].y);

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

{

l = 0;

r = m-1;

while (r-l > 1)

{

p = (r+l)>>1;

if (a[i].y > b[p].x)

r = p;

else

l = p;

}

if (a[i].y <= b[r].x)

l = r;

p = find(1, 0, N-1, 0, l, a[i].x);

if (p == -1)

{

printf("-1\n");

return 0;

}

ans += b[p-N].x;

update(p-N, -1);

}

printf("%lld\n", ans);

return 0;

}