#include <cstdio>

#include <algorithm>

using namespace std;

int n;

int d[300000];

struct mast

{

int h, k;

} m[100010];

bool operator < (mast a, mast b)

{

return a.h < b.h;

}

int val(int p)

{

int ret=0;

for (p += n-1; p; p>>=1)

ret += d[p];

return ret;

}

void add(int p, int a, int b, int s, int e)

{

if (a == s && b == e)

{

d[p]++;

return;

}

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

if (b <= l)

add(p<<1, a, b, s, l);

else

if (a > l)

add((p<<1)|1, a, b, l+1, e);

else

{

add(p<<1, a, l, s, l);

add((p<<1)|1, l+1, b, l+1, e);

}

}

int main()

{

long long ans=0;

int N, x, y, p, q, k, l, i;

scanf("%d", &N);

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

scanf("%d%d", &m[i].h, &m[i].k);

sort(m, m+N);

n = m[N-1].h;

while (n&(n-1))

n++;

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

{

if (m[i].k == m[i].h)

{

add(1, 1, m[i].k, 1, n);

continue;

}

x = val(m[i].h-m[i].k);

y = val(m[i].h-m[i].k+1);

if (x != y)

{

add(1, m[i].h-m[i].k+1, m[i].h, 1, n);

continue;

}

p = 1;

q = m[i].h-m[i].k;

while (q-p > 1)

{

k = (p+q)>>1;

if (val(k) != x)

p = k;

else

q = k;

}

if (val(p) == x)

k = p;

else

k = q;

p = m[i].h-m[i].k+1;

q = m[i].h;

while (q-p > 1)

{

l = (p+q)>>1;

if (val(l) != x)

q = l;

else

p = l;

}

if (val(q) == x)

l = q;

else

l = p;

if (l != m[i].h)

add(1, l+1, m[i].h, 1, n);

p = m[i].k-m[i].h+l;

add(1, k, k+p-1, 1, n);

}

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

{

x = val(i);

ans += x\*(1LL)\*(x-(1LL))/2;

}

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

return 0;

}