#include <cstdio>

#include <cmath>

#include <algorithm>

using namespace std;

long long x[1010], y[1010];

struct P

{

int p, q;

} e[1000010];

bool operator < (P a, P b)

{

return (x[a.p]-x[a.q])\*(x[a.p]-x[a.q])+(y[a.p]-y[a.q])\*(y[a.p]-y[a.q]) <

(x[b.p]-x[b.q])\*(x[b.p]-x[b.q])+(y[b.p]-y[b.q])\*(y[b.p]-y[b.q]);

}

int rank[1010], par[1010];

int find(int p)

{

if (p == par[p])

return p;

return p = par[p] = find(par[p]);

}

bool merge(int p, int q)

{

p = find(p);

q = find(q);

if (p == q)

return 0;

if (rank[p] > rank[q])

{

par[q] = p;

rank[p]++;

}

else

{

par[p] = q;

rank[q]++;

}

return 1;

}

int main()

{

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

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

double ans=0.0;

int n, m, p, q, i, j;

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

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

scanf("%lld%lld", x+i, y+i);

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

{

par[i] = i;

rank[i] = 1;

}

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

{

scanf("%d%d", &p, &q);

merge(p-1, q-1);

}

p = 0;

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

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

{

e[p].p = i;

e[p++].q = j;

}

sort(e, e+p);

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

if (merge(e[i].p, e[i].q))

{

if ((x[e[i].p]-x[e[i].q])\*(x[e[i].p]-x[e[i].q])+(y[e[i].p]-y[e[i].q])\*(y[e[i].p]-y[e[i].q]) < 0)

i = i;

ans += sqrt((x[e[i].p]-x[e[i].q])\*(x[e[i].p]-x[e[i].q])+(y[e[i].p]-y[e[i].q])\*(y[e[i].p]-y[e[i].q])+0.0);

}

printf("%.2lf\n", ans);

return 0;

}