#include

#include

using namespace std;

#define inf 10000

#define L(i) ((i<<1)-1)

#define R(i) (i<<1)

short M;

short n, m;

short ans = 0;

bool t[4010];

short w[4010][4010];

vector g[4010];

short x1[2010], y1[2010], x2[2010], y2[2010];

bool dfs(short p)

{

if (p == 2\*(n+m)+1)

{

ans++;

return 1;

}

short q, i;

if (g[p].size() == 0)

return 0;

for (i = g[p].size()-1; i >= 0; i--)

{

q = g[p][i];

if (!t[q] && w[p][q] > 0)

{

t[q] = 1;

if (dfs(q))

{

w[p][q]--;

w[q][p]++;

return 1;

}

}

}

return 0;

}

int main()

{

short p, i, j;

scanf("%hd", &n);

p = n;

for (m = 0, i = 1; i <= n; i++)

{

scanf("%hd%hd%hd%hd", x1+i, y1+i, x2+i, y2+i);

if (x1[i] == x2[i])

{

if (y1[i] > y2[i])

{

y1[i] ^= y2[i];

y2[i] ^= y1[i];

y1[i] ^= y2[i];

}

}

else

if (x1[i] > x2[i])

{

x1[i] ^= x2[i];

x2[i] ^= x1[i];

x1[i] ^= x2[i];

}

if (x1[i] != x2[i])

{

x1[p-m] = x1[i];

y1[p-m] = y1[i];

x2[p-m] = x2[i];

y2[p-m] = y2[i];

i--;

m++;

n--;

}

}

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

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

if (x1[j] <= x1[i] && x2[j] >= x1[i] &&

y1[i] <= y1[j] && y2[i] >= y1[j])

{

w[R(i)][L(j)] = w[R(j)][L(i)] = inf;

g[R(i)].push\_back(L(j));

g[L(j)].push\_back(R(i));

g[R(j)].push\_back(L(i));

g[L(i)].push\_back(R(j));

}

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

{

w[L(i)][R(i)] = 1;

g[L(i)].push\_back(R(i));

g[R(i)].push\_back(L(i));

if (i <= n)

{

w[0][L(i)] = inf;

g[0].push\_back(L(i));

g[L(i)].push\_back(0);

}

else

{

w[R(i)][2\*p+1] = inf;

g[R(i)].push\_back(2\*p+1);

g[2\*p+1].push\_back(R(i));

}

}

while (1)

{

memset(t, 0, sizeof(t));

t[0] = 1;

if (!dfs(0))

break;

}

printf("%hd\n", n+m-ans);

return 0;

}

/\*

6

0 0 0 1

0 1 1 1

1 1 1 2

1 2 2 2

2 2 2 0

2 0 0 0

4

0 0 2 0

1 1 1 -1

10 10 12 10

11 11 11 9

8

0 0 4 0

1 1 1 -1

2 2 2 -2

3 3 3 -3

10 10 14 10

11 11 11 9

12 12 12 8

13 13 13 7

8

0 -10 0 10

\*/