Wooden  
Có N thanh gỗ cho trước chiều dài L và khối lượng W. John muốn xếp chồng chúng lên sao cho được 1 khối cao nhất có thể nhưng thật khó vì khối xếp dưới phải có khối lượng và chiều dài lớn hơn hoặc bằng thanh gỗ ở trên.  
Vậy hãy cố gắng giúp john làm việc này nhé.: Làm sao để xếp được chồng gỗ lên cao nhất vậy?

Hỏi xếp được cao nhất với nhiều nhất bao nhiều thanh gỗ có thể ?

Example :  
2  
3  
2 2  
3 3  
4 4

4

15 17

10 9

12 13

12 15

Output: #1 3

#2 4

[#include](tg://search_hashtag?hashtag=include) <iostream>  
  
using namespace std;  
int ChieuDai[10];  
int KhoiLuong[10];  
int Visited[10];  
int SapXep[10];  
int N;  
int Answer;  
void Backtrack(int cnt){  
if(Answer == N)  
return;  
if(cnt > Answer){  
Answer = cnt;  
}  
  
for(int i = 0; i < N; i++){  
if(Visited[i] == 0){  
if(KhoiLuong[i] <= KhoiLuong[SapXep[cnt-1]] && ChieuDai[i] <= ChieuDai[SapXep[cnt-1]]){  
SapXep[cnt] = i;  
Visited[i] = 1;  
Backtrack(cnt+1);  
Visited[i] = 0;  
SapXep[cnt] = 0;  
}  
}  
}  
  
}  
int main()  
{  
int tc,TestCase, output = 0;  
  
freopen("Text.txt", "r", stdin);  
freopen("Text1.txt", "w", stdout);  
cin>>TestCase;  
  
for(tc=0; tc<TestCase; tc++)  
{  
output = 0;  
Answer = 0;  
  
cin >> N;  
for(int i = 0; i < N; i++){  
cin >> ChieuDai[i] >> KhoiLuong[i];  
Visited[i] = 0;  
}  
  
  
////////////////////////////////  
for(int i = 0; i < N; i++){  
Visited[i] = 1;  
SapXep[0] = i;   
Backtrack(1);  
Visited[i] = 0;  
}  
  
  
////////////////////////////////  
  
  
  
cout<< "#" << tc+1 <<" "<< Answer << endl;  
}  
  
return 0;  
}

Input

Hỏi xếp được bao nhiêu thanh gỗ nhất có thể ?

50  
3  
11 50  
68 18  
19 38  
3  
59 55  
12 45  
8 89  
3  
90 87  
44 76  
66 87  
3  
15 79  
52 91  
67 68  
3  
62 15  
61 10  
97 10  
4  
92 25  
45 89  
45 96  
59 98  
4  
81 46  
93 49  
55 65  
43 35  
4  
48 79  
81 76  
69 6  
16 66  
4  
78 42  
79 72  
24 88  
69 41  
4  
90 48  
94 55  
50 12  
36 40  
4  
60 40  
64 93  
53 99  
54 84  
4  
21 50  
45 45  
48 46  
47 40  
4  
80 57  
49 43  
5 62  
55 81  
5  
30 7  
50 72  
93 67  
75 64  
7 52  
5  
14 38  
37 53  
20 12  
39 5  
73 40  
5  
11 5  
55 91  
56 5  
43 41  
75 67  
5  
91 22  
46 33  
83 92  
87 20  
60 19  
5  
86 58  
40 75  
8 56  
47 79  
93 94  
5  
61 72  
44 54  
60 41  
82 27  
7 65  
6  
56 10  
83 62  
73 16  
22 30  
33 65  
49 9  
6  
94 75  
84 58  
77 92  
47 18  
51 89  
23 80  
6  
70 46  
13 38  
47 65  
5 51  
55 95  
80 30  
6  
85 76  
7 86  
51 29  
97 93  
65 22  
92 34  
6  
62 97  
18 77  
13 91  
69 95  
20 97  
63 62  
7  
98 53  
70 98  
23 18  
47 57  
37 70  
24 82  
35 45  
7  
43 14  
44 41  
81 66  
75 80  
59 41  
24 76  
19 99  
7  
34 55  
65 75  
14 85  
57 55  
56 56  
65 78  
51 87  
7  
99 49  
69 50  
92 65  
17 7  
45 7  
78 69  
57 51  
7  
96 19  
5 56  
12 12  
89 97  
50 95  
66 17  
9 16  
7  
13 45  
60 77  
7 60  
48 78  
27 63  
63 37  
10 25  
7  
48 64  
72 22  
55 94  
26 54  
95 38  
92 97  
89 9  
8  
73 41  
33 77  
91 96  
46 53  
91 56  
41 96  
64 14  
88 75  
8  
97 17  
43 53  
29 96  
39 63  
84 64  
37 99  
73 96  
90 16  
8  
72 44  
5 64  
98 28  
27 98  
79 84  
27 73  
58 11  
64 21  
8  
42 17  
91 37  
39 49  
28 38  
76 24  
47 85  
34 22  
8 20  
8  
75 47  
50 48  
21 63  
10 39  
35 90  
9 45  
41 60  
67 77  
8  
25 13  
56 59  
68 60  
11 48  
78 17  
38 29  
89 28  
61 77  
8  
37 35  
32 87  
76 75  
36 97  
99 7  
85 20  
13 49  
36 75  
9  
72 8  
34 99  
55 54  
13 57  
27 54  
19 36  
57 22  
51 95  
43 69  
9  
20 38  
49 7  
87 99  
31 83  
26 88  
11 42  
99 59  
70 11  
58 17  
9  
83 23  
96 13  
27 87  
33 30  
23 89  
93 66  
28 43  
48 40  
18 91  
9  
14 76  
36 86  
51 41  
62 74  
39 80  
99 77  
48 46  
64 87  
19 49  
9  
38 20  
61 51  
43 22  
71 7  
87 32  
49 94  
5 67  
53 8  
9 33  
9  
74 53  
61 36  
24 50  
15 47  
52 80  
23 51  
38 75  
16 13  
95 45  
10  
88 42  
20 85  
92 6  
73 75  
15 62  
96 86  
98 86  
7 64  
19 64  
9 50  
10  
33 43  
6 88  
74 93  
6 63  
20 30  
93 13  
55 77  
44 12  
8 7  
12 68  
10  
66 29  
84 11  
93 66  
70 79  
28 52  
76 71  
10 20  
44 13  
50 80  
44 30  
10  
17 49  
26 82  
40 69  
55 90  
19 78  
63 10  
53 37  
36 49  
49 19  
52 75  
10  
18 42  
98 94  
38 42  
25 62  
8 23  
36 35  
75 95  
79 89  
33 91  
97 96  
10  
21 58  
60 41  
15 96  
36 31  
23 49  
41 28  
81 8  
74 12  
53 54  
13 17

Output:

#1 2  
#2 2  
#3 3  
#4 2  
#5 2  
#6 3  
#7 3  
#8 3  
#9 3  
#10 3  
#11 3  
#12 2  
#13 2  
#14 5  
#15 4  
#16 3  
#17 4  
#18 5  
#19 2  
#20 4  
#21 4  
#22 3  
#23 3  
#24 3  
#25 5  
#26 4  
#27 4  
#28 4  
#29 4  
#30 5  
#31 4  
#32 4  
#33 3  
#34 5  
#35 5  
#36 4  
#37 5  
#38 4  
#39 4  
#40 5  
#41 3  
#42 5  
#43 5  
#44 6  
#45 7  
#46 5  
#47 6  
#48 7  
#49 6  
#50 3

package luyende;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.util.Scanner;

/\*

\* Cách 1 là backTracking, cách 2 là làm quy hoạch động giống như bài dãy con tăng dài nhất (mảng 1 chiều) thì mình làm ở đấy là mảng 2 chiều

\*/

public class Woonden2 {

Scanner sc = new Scanner(System.in);

int t,n;

int arr[][], L[];

void init() {

L = new int[n+1];

arr = new int[n+1][2];

}

void solution() {

t = sc.nextInt();

for(int tc=1; tc<=t; tc++) {

n = sc.nextInt();

init();

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

arr[i][0] = sc.nextInt();

arr[i][1] = sc.nextInt();

}

// System.out.println("arr");

// for(int i=0; i<n; i++) {

// System.out.println(arr[i][0]+" "+arr[i][1]);

// }

for(int i=0; i<n-1; i++) {

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

if(arr[i][0] < arr[j][0]) {

int temp = arr[i][0];

arr[i][0] = arr[j][0];

arr[j][0] = temp;

temp = arr[i][1];

arr[i][1] = arr[j][1];

arr[j][1] = temp;

}

}

}

// System.out.println("arr");

// for(int i=0; i<n; i++) {

// System.out.println(arr[i][0]+" "+arr[i][1]);

// }

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

L[i] = 1;

}

for(int i=0; i<n-1; i++) {

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

if(arr[i][0] >= arr[j][0] && arr[i][1] >= arr[j][1]) {

if((L[i] + 1) >= L[j]) {

L[j] = L[i]+1;

}

}

}

}

// System.out.println("L[]");

// for(int i=0; i<n; i++) {

// System.out.print(L[i]+" ");

// }

for(int i=0; i<n-1; i++) {

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

if(L[i] < L[j]) {

int temp = L[i];

L[i] = L[j];

L[j] = temp;

}

}

}

// System.out.println("L[]");

// for(int i=0; i<n; i++) {

// System.out.print(L[i]+" ");

// }

System.out.println("#"+tc+" "+L[0]);

}

}

public static void main(String[] args) throws Exception {

System.setIn(new FileInputStream("Wooden.txt"));

Woonden2 w = new Woonden2();

w.solution();

}

}