```
import java.io.*;
import java.util.*;
import java.lang.*;


public class Main
{
    void solve() throws Exception
    {

        int N = ni();
        List<int[]> breakfasts = new ArrayList<>();
        int[] breakfast;
        for(int i = 0; i< N; i++) {
            breakfast = new int[]{ni(), ni(), ni(), ni()};
            breakfasts.add(breakfast);
        }
        breakfasts.sort((o1, o2) -> {
            int left;
            int right;
            if (o1[1] == o2[1]){
                if (o1[2] == o2[2]){
                    left = o1[3];
                    right = o2[3];
                } else {
                    left = o1[2];
                    right = o2[2];
                }
            } else {
                left = o1[1];
                right = o2[1];
            }
            return Integer.compare(left, right);
        });


        int[] LIS = new int[1+N];

        int sz = 0, Lmax = -1;
        for(int i = 0; i< N; i++){
            int x = breakfasts.get(i)[0], L = breakfasts.get(i)[2], R = breakfasts.get(i)[3];

            if(R <= Lmax){
                pn(-1);
                return;
            }
            while(sz > 0 && LIS[sz] >= R)sz--;
            if(x <= Lmax)continue;


            int lo = 1, hi = sz;
            while (lo < hi){
                int mid = lo + (hi-lo)/2;
                if(LIS[mid] >= x)hi = mid;
                else lo = mid+1;
        }
        if(hi > 0 && LIS[hi] >= x)LIS[hi] = x;
        else LIS[++sz] = x;

        Lmax = Math.max(Lmax, L);
    }
    pn(N-sz);
}
static boolean multipleTC = true;
FastReader in;
PrintWriter out;
void run() throws Exception{
    in = new FastReader();
    out = new PrintWriter(System.out);

    int T = (multipleTC)? ni(): 1;
    for(int t = 1; t<= T; t++)
        solve();
    out.flush();
    out.close();
}

public static void main(String[] args) throws Exception{
    new Main().run();
}

int ni() throws Exception{
    return Integer.parseInt(in.next());
}

void pn(Object o){
    out.println(o);
}

static class FastReader{
    BufferedReader br;
    StringTokenizer st;
    public FastReader(){
        br = new BufferedReader(new InputStreamReader(System.in));
    }

    public FastReader(String s) throws Exception{
        br = new BufferedReader(new FileReader(s));
    }

    String next() throws Exception{
        while (st == null || !st.hasMoreElements()){
            try{
                st = new StringTokenizer(br.readLine());
            }catch (IOException e){
                throw new Exception(e.toString());
            }
        }
        return st.nextToken();
    }
```

```
        String nextLine() throws Exception{
            String str = "";
            try{
                str = br.readLine();
            }catch (IOException e){
                throw new Exception(e.toString());
            }
            return str;
        }
    }
}
```