CF1379C

有 m 种花，每种花第一次选有 a_i 的快乐值，之后每次有 b_i 的快乐值。要选 n 朵花，并最大化快乐值

直接贪心似乎没什么思路，考虑比较暴力的做法：枚举第 i 种花，把所有 $a_j \ge b_i$ 的其他花先取一次，剩下全部取第 i 种花。

```
#include<bits/stdc++.h>
#define ALL(x) (x).begin(),(x).end()
#define ll long long
#define db double
#define ull unsigned long long
#define pii_ pair<int,int>
#define mp_ make_pair
#define pb push_back
#define fi first
#define se second
#define rep(i,a,b) for(int i=(a);i<=(b);i++)
#define per(i,a,b) for(int i=(a);i>=(b);i--)
#define show1(a) cout<<#a<<" = "<<a<<endl
#define show2(a,b) cout<<#a<<" = "<<a<<"; "<<#b<<" = "<<b<<endl
using namespace std;
const ll INF = 1LL<<60;
const int inf = 1<<30;
const int maxn = 1e5+5;
inline void fastio() {ios::sync_with_stdio(false);cin.tie(0);cout.tie(0);}

int a[maxn],b[maxn];

int main()
{
    fastio();
    int _,n,m;
    for(cin>>_;_;_--){
        cin>>n>>m;
        ll ans = 0;
        vector<ll> vec,suf;
        rep(i,1,m){
            cin>>a[i]>>b[i];
            vec.pb(a[i]);
        }
        sort(ALL(vec));
        per(i,m-1,0){
            if(i==m-1) suf.pb(vec[i]);
            else suf.pb(suf.back() + vec[i]);
            //show1(suf.back());
        }
        rep(i,1,m){
            int pos = lower_bound(ALL(vec),b[i]) - vec.begin();
```

```
            int r = m - pos;
            //show2(b[i],r);
            if(r>=n){
                //show1(suf[n-1]);
                ans = max(ans,suf[n-1]);
            }else{
                ll tmp = r?suf[r-1]:0;
                if(a[i] >= b[i]){
                    tmp += (ll) b[i] * (n-r);
                }else{
                    tmp += (ll)a[i] + (ll)b[i] * (n-r-1);
                }
                //show1(tmp);
                ans = max(ans,tmp);
            }
        }
        cout<<ans<<endl;
    }
    return 0;
}
```