[HDU 6119 小小粉丝度度熊（Two pointers）](https://www.cnblogs.com/forever97/p/hdu6119.html)

【题目链接】 <http://acm.hdu.edu.cn/showproblem.php?pid=6119>

【题目大意】

　　给出一些签到区间和一些补签卡，问可以创造的最长连续签到区间

【题解】

　　如果我们知道选定的最左和最右的签到区间，  
　　我们就可以计算出需要补多少的补签卡，如果数量小于等于给定数量，  
　　那么这个左右可以用来更新答案。所以尺取法就显而易见了。

【代码】

1. #include <cstdio>
2. #include <algorithm>
3. using namespace std;
4. const int N=100010;
5. struct data{int l,r;}p[N];
6. int l[N],r[N],cnt;
7. bool cmp(data a,data b){
8. if(a.l==b.l)return a.r<b.r;
9. return a.l<b.l;
10. }
11. int n,m;
12. int main(){
13. while(~scanf("%d%d",&n,&m)){
14. for(int i=1;i<=n;i++)scanf("%d%d",&p[i].l,&p[i].r);
15. sort(p+1,p+n+1,cmp); cnt=1;
16. l[cnt]=p[1].l,r[cnt]=p[1].r;
17. for(int i=2;i<=n;i++){
18. if(p[i].l-1>r[cnt]){l[++cnt]=p[i].l;r[cnt]=p[i].r;}
19. r[cnt]=max(r[cnt],p[i].r);
20. }
21. int ans=r[1]-l[1]+1+m,gap=0,lft=1;
22. for(int rt=2;rt<=cnt;rt++){
23. gap+=l[rt]-1-r[rt-1];
24. while(gap>m){lft++;gap-=l[lft]-1-r[lft-1];}
25. ans=max(ans,r[rt]-l[lft]+1+m-gap);
26. }printf("%d\n",ans);
27. }return 0;
28. }