bool Halfplane\_intersection(int n, Line \*hp, Point \*p) {
  
 if(n < 3) return false;
  
 sort(hp, hp + n, cmp);
  
 Halfplane\_unique(n, hp);
  
 st = 0; ed = 1;
  
 que[0] = 0; que[1] = 1;
  
 if(parallel(hp[0], hp[1])) return false;
  
 Calc\_intersection(hp[0], hp[1], p[1]);
  
 for(int i = 2; i < n; i++) {
  
 while(st < ed &&
  
 sgn((hp[i].t-hp[i].s) \* (p[ ed ]-hp[i].s)) < 0)
  
 ed--;
  
 while(st < ed &&
  
 sgn((hp[i].t-hp[i].s) \* (p[st+1]-hp[i].s)) < 0)
  
 st++;
  
 que[++ed] = i;
  
 assert(ed >= 1);
  
 if(parallel(hp[i], hp[que[ed - 1]])) return false;
  
 Calc\_intersection(hp[i], hp[que[ed - 1]], p[ed]);
  
 }
  
 while(st < ed &&
  
 sgn((hp[que[st]].t-hp[que[st]].s) \* (p[ ed ]-hp[que[st]].s)) < 0)
  
 ed--;
  
 while(st < ed &&
  
 sgn((hp[que[ed]].t-hp[que[ed]].s) \* (p[st+1]-hp[que[ed]].s)) < 0)
  
 st++;
  
 if(st + 1 >= ed) return false;
  
 return true;
  
}