

## Algorithmique TP 22 : Algorithmique géométrique

### Chasse à l'ours

La chasse est ouverte :  $n$  chasseurs partent à la poursuite de  $n$  ours. C'est une chasse moderne (!) où le chasseur utilise un fusil à laser. C'est une chasse organisée où tous les chasseurs font feu au même instant. Le seul inconvénient de cette modernité est qu'il est interdit de croiser deux tirs (risque de désintégrations des deux tireurs fautifs).

On s'intéresse au cas où les chasseurs et les ours sont bien élevés : d'une part ils ne bougent pas, d'autre part ils sont répartis (dans la province Cartésienne) de telle façon qu'il n'y ait pas d'alignements de trois individus, chasseur ou ours.

1. Prouver qu'il existe une droite passant par un chasseur et un ours tel que le nombre d'ours d'un côté de la droite soit égal au nombre de chasseurs de ce même côté. Donner un algorithme pour trouver une telle droite, si possible de complexité  $O(n \log n)$ . Indication : considérer l'individu (ours ou chasseur) le plus au sud et sélectionner son partenaire (!) en balayant les autres individus par angle croissant (de l'est vers l'ouest).
2. Proposer un algorithme qui associe un ours à chaque chasseur afin que la condition de non-croisement des tirs soit respectée (penser récursivité).
3. Implémenter l'algorithme en OCaml. On pourra utiliser la fonction de tri `List.sort` et illustrer avec la fenêtre graphique.

