

Algorithmique

Novembre 2015 – Un problème de jeu de cartes

Problème :

On considère un jeu de cartes à deux joueurs. Au début du jeu, on dispose une suite de n cartes c_1, \dots, c_n sur une table (face visible). Chaque carte a une valeur entière : on note v_i la valeur de la carte c_i , dans la suite on utilisera seulement v_i pour désigner la carte i . A chaque tour, le joueur 1 choisit **une** des deux cartes situées à une extrémité (gauche ou droite) de la rangée. Le joueur 2 fait ensuite de même. Le jeu s'arrête lorsque toutes les cartes ont été prises par les deux joueurs. Le gagnant est celui dont la somme des valeurs de ses cartes est la plus grande (et il y a match nul en cas d'égalité).

Exemple : on part de la suite de valeurs suivante $[4;2;8;12;45;2;6;9;0;6]$. Le joueur 1 prend la carte de gauche (celle de valeur 4), le joueur 2 prend celle de droite (donc de valeur 6). On a alors la suite $[2;8;12;45;2;6;9;0]$. Puis le joueur 1 prend 2 et le joueur 2 prend 8 (on arrive à $[12;45;2;6;9;0]$), puis le joueur 1 prend 0, le joueur 2 prend 9 (on arrive à $[12,45,2;6]$), le joueur 1 prend 6 et le joueur 2 prend 12 (on arrive à $[45;2]$) et le joueur 1 prend 45 et le joueur 2 prend 2. Le joueur 1 a donc récupéré $4+2+0+6+45=57$, tandis que le joueur 2 a pris $6+8+9+12+2=37$. C'est donc le joueur 1 qui gagne cette partie.

Notons qu'à chaque tour, la position du jeu est définie par une paire (i, j) avec $1 \leq i \leq j \leq n$ et une information indiquant quel joueur a le trait (*ie* le prochain qui doit jouer) : la carte la plus à gauche est c_i et celle la plus à droite est c_j . Une telle paire est appelée une **configuration** du jeu. La configuration initiale est $(1, n)$.

L'objectif de cet exercice est de calculer la (ou une) *meilleure stratégie* (c'est-à-dire la meilleure manière de jouer) possible pour le joueur 1.

1. Montrer que la stratégie gloutonne « choisir toujours la carte ayant la plus grande valeur » n'est pas toujours optimale.
2. Proposer un algorithme en $O(n^2)$ pour calculer une table $\text{TV}[-, -]$ de dimension $n \times n$ telle que pour toute configuration (i, j) , $\text{TV}[i, j]$ contient la valeur optimale que peut **garantir** le joueur 1 (*ie* même lorsque le joueur 2 joue le mieux possible). A quelle famille, cet algorithme appartient-il ? Quelle est sa complexité ?
3. Proposer un algorithme qui, à partir de cette table $\text{TV}[-, -]$, indiquera dans chaque configuration un meilleur choix pour le joueur 1. Quelle est sa complexité ?
4. Est-ce que le joueur qui commence a une stratégie gagnante ? On pourra distinguer les cas n pair et n impair.