

Le jeu de Marienbad

Ce jeu, qui apparaît dans le film *l'année dernière à Marienbad*, s'appelle aussi jeu de Nim. Connue depuis longtemps, il s'appelle *fan tan* en Chine et *tiouk tiouk* en Afrique. Voici la règle du jeu. Au départ, des allumettes sont réparties en plusieurs tas (ou piles). Deux personnes vont jouer à tour de rôle. Chacune peut enlever un nombre quelconque d'allumettes (au moins une) dans un et un seul tas de son choix. Le joueur gagnant est celui qui enlève les dernières allumettes.

Sans entrer dans le détail de la théorie, voici la stratégie permettant au joueur qui joue en premier de gagner, sauf cas exceptionnels.

Prenons un exemple avec quatre piles comptant 3, 5, 6, et 7 allumettes au début. On commence par convertir ces quatre nombres en binaire (on choisit l'écriture suivant les poids décroissants), puis on fait la somme des chiffres colonne par colonne en la ramenant modulo 2 (il s'agit en fait d'un XOR). Le joueur qui gagne est celui qui après avoir joué obtient une somme nulle à chaque fois. Au départ, on a :

tas 0	3	011
tas 1	5	101
tas 2	6	110
tas 3	7	<u>111</u>
somme		111

Le premier joueur s'arrange pour que cette somme devienne nulle. Comment faire ? Il commence par considérer le 1 le plus à gauche dans la somme, ici il est dans la première colonne. Il choisit alors un tas qui a aussi un 1 dans cette même colonne, par exemple ici le tas 1. Dans ce tas 1, il remplace ce 1 par 0, ce qui signifie qu'il enlève 4 allumettes et le tas 1 devient 001. La somme est alors 011, mais pas encore 000. Pour que le 1 de la deuxième colonne de la somme devienne 0, il faut remplacer le 0 du tas 1 par un 1. On fait de même avec le 1 de la somme dans la troisième colonne, en remplaçant le 1 du tas 1 dans la même colonne par un 0. Finalement le joueur enlève $4 - 2 + 1 = 3$ allumettes dans le tas 1. On obtient alors :

tas 0	3	011
tas 1	5	010
tas 2	6	110
tas 3	7	<u>111</u>
somme		000

Quoi que fasse maintenant le deuxième joueur, il ne pourra jamais obtenir une somme formée uniquement de 0. Après cela, lorsque le premier joueur va rejouer, il pourra, lui, retrouver une somme de 0, en appliquant la même méthode que précédemment. Et ainsi de suite jusqu'au dénouement, où chaque tas n'est formé que de 0 et la somme aussi, après le dernier coup du premier joueur.

Résumons la méthode : on cherche le 1 le plus à gauche dans la somme, puis on prend le tas qui contient un 1 dans la même colonne, et on le remplace par 0 (comme on prend la colonne la plus à gauche, il s'agit d'un poids fort, et le remplacement du 1 par 0

dans le tas correspond à une diminution du nombre des allumettes, quoi qu'il se passe ensuite). Puis on prend à tour de rôle tous les autres 1 de la somme, et dans les colonnes correspondantes du tas concerné, on remplace soit un 0 par 1, soit un 1 par 0. La somme devient alors une succession de 0. Comme la fin du jeu correspond aussi à une succession de 0, le joueur qui arrive à avoir une succession de 0 à chaque fois qu'il joue va gagner.

Si la personne jouant en premier connaît cette stratégie, elle est sûre de gagner, sauf si exceptionnellement la répartition initiale des allumettes donne une somme nulle. Par exemple si l'on prend des piles de 2, 3, 6 et 7 allumettes, on a une somme nulle. Dans ce cas le premier joueur ne pourra pas obtenir une somme nulle après avoir joué, et c'est le deuxième joueur, du moins s'il connaît la stratégie, qui va gagner.

Dans le programme de jeu automatisé qui suit, on a choisi un scénario simplifié : le premier joueur est en fait la machine, et elle connaît la stratégie. Quant au deuxième joueur, il joue aussi automatiquement, mais ne connaît pas la stratégie, il se contente d'enlever une allumette dans une pile choisie au hasard. Dans ces conditions, la machine est sûre de gagner si la somme initiale n'est pas nulle. Et si la somme est nulle, la machine joue modestement en enlevant une allumette dans la plus grosse pile, ce qui donne une somme non nulle, mais comme le second joueur ne connaît pas la stratégie, c'est encore la machine qui a quasiment toutes les chances de gagner car le second joueur ne va pas obtenir une somme nulle au premier coup, et la machine va l'obtenir, elle, au second coup.

Programme

```

...
#define N 5  /* N est le nombre de tas ou de piles */
void sommexor(void); /* cette fonction calcule la somme modulo 2 */
void afficherpiles(void);
int p[N],r[N][10],s[10],nbchiffres;

int main()
{ int puis2,i,j,cg1,ichanger,cumul,oldp,fini,choix,pilechoisie,numero;
  printf("Nombre de piles: %d\n\n",N);
  p[0]=1; p[1]=3; p[2]=6; p[3]=8; p[4]=12; /* donnée des N piles */

  puis2=1;nbchiffres=0; /* nbchiffres est la longueur de chaque nombre en binaire, en
  while( puis2<=p[N-1]) { nbchiffres++; puis2*=2;} /* supposant que le dernière pile est la
  sommexor();afficherpiles(); getch(); /* plus grosse */

  for(;;)
  { choix=1;
    for(j=0;j<nbchiffres;j++) if(s[j]!=0) { choix=0; break;}
    if(choix==1) /* on a choix = 1 lorsque la somme est nulle */
    { for(i=N-1;i>=0;i--) if(p[i]!=0) { pilechoisie=i; break;}
      p[pilechoisie]--;
      printf("\n\n*****LA MACHINE JOUE: elle enleve un jeton a la pile %d",
      pilechoisie);
      sommexor();afficherpiles(); getch();
    }
    else if(choix==0)
    { for(j=0;j<nbchiffres;j++) if (s[j]==1) { cg1=j; break;}
  }
}

```

```

for(i=0;i<N;i++) if(r[i][cg1]==1) {iachanger=i;break;}
oldp=p[iachanger];
r[iachanger][cg1]=0;
for(j=cg1+1; j<nbchiffres;j++)
if (s[j]==1) r[iachanger][j]=(r[iachanger][j]+1)%2;
puis2=1; cumul=0;
for(j=nbchiffres-1; j>=0; j--) { cumul+=puis2*r[iachanger][j]; puis2*=2;}
p[iachanger]=cumul;
printf("\n\nLA MACHINE JOUE: ");
printf("elle enleve %d allumette(s) dans la pile %d",oldp-p[iachanger], iachanger);
sommexor(); afficherpiles(); getch(); printf("\n");
}
fini=1;
for(i=0;i<N;i++) if (p[i]!=0) {fini=0; break;}
if(fini==1) { printf("\n\n**** LA MACHINE A GAGNE ****\n\n"); getch(); break;}
do numero=rand()%N; while (p[numero]==0);
p[numero]--;
printf("\n\nLE JOUEUR JOUE: ");
printf("il enleve 1 allumette dans la pile %d",numero);
sommexor();afficherpiles(); getch();
}
return 0;
}

void sommexor(void)
{ int i,j,q;
  for(i=0;i<N;i++) /* conversion des nombres en binaire décroissant */
  { q=p[i]; for(j=0;j<nbchiffres;j++) { r[i][nbchiffres-1-j]=q%2; q=q/2;} }
  for(j=0;j<nbchiffres;j++) { s[j]=0; for(i=0;i<N;i++) s[j]+=r[i][j];s[j]=s[j]%2;}
}
void afficherpiles(void)
{ int i; for(i=0;i<N;i++) printf("\npile %d : %d ",i,p[i]); printf("\n"); }

```

Exercice : réaliser une version où les tas d'allumettes sont visualisés sur l'écran, avec le joueur extérieur qui joue contre la machine, et qui peut enlever les allumettes de son choix dans une des piles en cliquant dessus.