

TP d'algorithmique appliquée : le jeu des allumettes

Bob aime jouer au jeu des allumettes. Dans ce jeu à deux participants, on débute avec 13 allumettes. Chacun à son tour, les joueurs retirent entre une et trois allumettes. Celui qui retire la dernière allumette a perdu.

1.1 Allumettes pour débutants

Bob (le joueur électronique) est le joueur 0, et son adversaire humain est le joueur 1 (Remarquons au passage que $1 - 0 = 1$ et $1 - 1 = 0$).

Pour faciliter la tâche à son adversaire, Bob joue complètement au hasard.

Exercice 1. Dans un premier temps, Bob propose de commencer. Écrire un algorithme du jeu des allumettes dans lequel :

- Bob commence.
- Bob retire un nombre aléatoire d'allumettes compris entre 1 et 3 bornes incluses (*indice* : utiliser les fonctions `floor()` et `random()` d'algobox). **Attention** : Bob ne peut pas retirer plus d'allumettes qu'il n'en reste. L'algorithme affiche alors le nombre d'allumettes restantes.
- Bob propose ensuite à son adversaire humain de choisir le nombre d'allumettes qu'il veut retirer. **Attention** : le nombre d'allumettes choisi par l'adversaire de Bob doit aussi être compris entre 1 et 3, et il ne peut pas retirer plus d'allumettes qu'il n'en reste. Tant que l'adversaire de Bob choisi un nombre d'allumettes qui ne convient pas, l'algorithme doit redemander le nombre d'allumettes à retirer.
- Si le nombre est correct, ces allumettes sont retirées et l'algorithme affiche le nombre d'allumettes restantes.
- Et ainsi de suite jusqu'à qu'il ne reste plus qu'une allumette. L'algorithme affiche alors le nom du gagnant (ou du perdant !).

Exercice 2. Cette fois-ci, Bob propose à son adversaire de choisir qui commence. Modifier l'algorithme précédent en conséquence.

1.2 Allumettes pour experts

Maintenant, Bob en a marre de perdre. Cette fois-ci c'est sûr, il va écraser son adversaire.

Exercice 3.

1. Écrire la division euclidienne de 13 par 4.
2. Montrer que si celui qui joue en deuxième enlève systématiquement 4 moins le nombre d'allumettes retirées par celui qui joue en premier, alors il gagne forcément.
3. Il reste n allumettes (où n est un entier strictement positif quelconque). Si $(n-1)$ n'est pas un multiple de 4, que doit-on jouer pour être sûr de gagner ?

Exercice 4. Reprendre l'algorithme de la section 1.1 en changeant le comportement de Bob (Bob fait tout pour gagner).

Gros indice : en notant n le nombre d'allumettes restantes, si $(n-1) \% 4 == r$:

1. Si $r > 0$, alors Bob retire r allumettes.
2. Si $r = 0$, alors Bob joue au hasard.

Code de l'algorithme

```
1  VARIABLES
2    a EST_DU_TYPE NOMBRE //nombre d'allumettes restantes
3    r EST_DU_TYPE NOMBRE // allumettes retirées
4    j EST_DU_TYPE NOMBRE // joueur (0 pour Bob, 1 pour l'adversaire)
5  DEBUT_ALGORITHME
6    a PREND_LA_VALEUR 13 // on commence avec 13 allumettes
7    j PREND_LA_VALEUR 1
8    TANT_QUE (a>1) FAIRE // boucle tant qu'il reste plus d'1 allumette
9      DEBUT_TANT_QUE
10     j PREND_LA_VALEUR -j+1
11     r PREND_LA_VALEUR 0
12     SI (j==0) ALORS
13       DEBUT_SI
14         TANT_QUE (r<=0 ou r>=a) FAIRE
15           DEBUT_TANT_QUE
16             r PREND_LA_VALEUR floor(random()*3+1)
17           FIN_TANT_QUE
18         a PREND_LA_VALEUR a-r
19         AFFICHER "Bob a retiré "
20         AFFICHER r
21         AFFICHER " allumette(s). "
22       FIN_SI
23     SINON
24       DEBUT_SINON
25         AFFICHER "Combien d'allumettes voulez-vous retirer ? "
26         r PREND_LA_VALEUR 0
27         TANT_QUE (r<=0 ou r>3 ou r>=a) FAIRE
28           DEBUT_TANT_QUE
29             LIRE r
30           FIN_TANT_QUE
31         a PREND_LA_VALEUR a-r
32       FIN_SINON
33     AFFICHER "Il reste "
34     AFFICHER a
35     AFFICHER " allumette(s). "
36     FIN_TANT_QUE
37   SI (j==0) ALORS
38     DEBUT_SI
39     AFFICHER "Vous avez perdu ! "
40   FIN_SI
41   SINON
42     DEBUT_SINON
43     AFFICHER "Bob a perdu !"
44   FIN_SINON
45  FIN_ALGORITHME
```

```
Bob a retiré 1 allumette(s).
Il reste 12 allumette(s).
Combien d'allumettes voulez-vous retirer ?
Entrer r : 3
Il reste 9 allumette(s).
Bob a retiré 1 allumette(s).
Il reste 8 allumette(s).
Combien d'allumettes voulez-vous retirer ?
Entrer r : 3
Il reste 5 allumette(s).
Bob a retiré 1 allumette(s).
Il reste 4 allumette(s).
Combien d'allumettes voulez-vous retirer ?
Entrer r : 3
Il reste 1 allumette(s).
Bob a perdu !
***Algorithme terminé***
```

Exercice 2.

```
7      j PREND_LA_VALEUR 2
8      TANT_QUE (j<0 ou j>1) FAIRE
9          DEBUT_TANT_QUE
10         AFFICHER "Qui commence (0 : moi (Bob), 1 : vous) ? "
11         LIRE j
12         FIN_TANT_QUE
13     j PREND_LA_VALEUR -j+1
```

Exercice 3.

1. La division entière de 13 par 4 est égale à 3, il reste 1. Cela s'écrit en algobox : $13\%4=1$
2. Enlever 4 moins le nombre d'allumettes retirées par le joueur précédent ne modifie pas le reste de la division euclidienne, mais modifie son quotient d'une unité. On passe de 13 à 9, puis à 5, puis à 1. A la dernière étape, il restera forcément une seule allumette.
3. Un nombre r tel que $(n-r-1)$ soit un multiple de 4.

Exercice 4.

```
Code de l'algorithme
1      ...
12     SI (j==0) ALORS
13         DEBUT_SI
14         SI ((a-1)%4==0) ALORS
15             DEBUT_SI
16             TANT_QUE (r<1 OU r>=a) FAIRE
17                 DEBUT_TANT_QUE
18                 r PREND_LA_VALEUR floor(random()*3+1)
19                 FIN_TANT_QUE
20             FIN_SI
21         SINON
22             DEBUT_SINON
23             r PREND_LA_VALEUR (a-1)%4
24             FIN_SINON
25         a PREND_LA_VALEUR a-r
26         AFFICHER "Bob a retiré "
27         AFFICHER r
28         AFFICHER " allumette(s). "
29         FIN_SI
30     SINON
31     ...
52 FIN_ALGORITHME
```