Tom le Robot

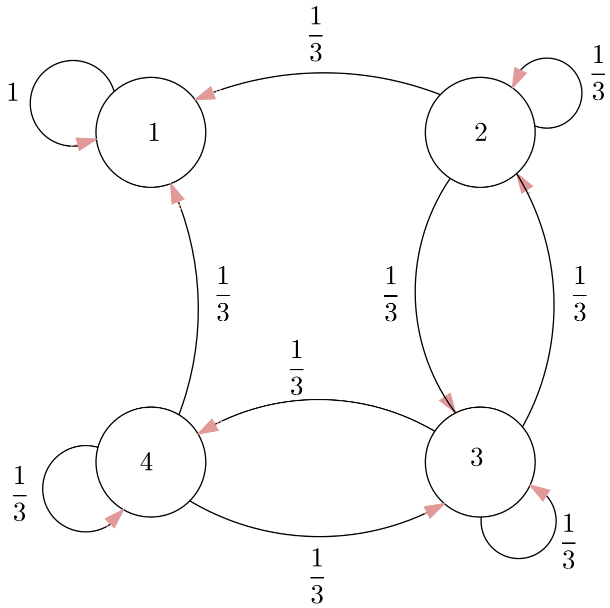
Très intéressant ce sujet mais… pourquoi est-il dans la catégorie « obligatoire » ? Si on le regarde d'un œil différent, c'est en fait un sujet de spécialité ! Les auteurs de ce sujet de BAC auraient pu choisir de traiter le même exercice, mais sous deux approches différentes.

Voyons cela de plus près !

Il s'agit évidemment d'une marche aléatoire. Dans le cadre de l'article, cette marche possède quatre états :

- ❶ : être dans l'eau ;
- ❷ : être un pas sur la gauche du milieu du pont ;
- ❸ : être au milieu du pont ;
- ❹ : être un pas sur la droite du milieu du pont ;

D'où le graphe probabiliste suivant :



La matrice de transition associée est :

$$M = \begin{pmatrix} 1 & \frac{1}{3} & 0 & \frac{1}{3} \\ 0 & \frac{1}{3} & \frac{1}{3} & 0 \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & 0 & \frac{1}{3} & \frac{1}{3} \end{pmatrix}$$

où pour tous $i, j = 1 \dots 4$, m_{ij} représente la probabilité de passer de l'état j à l'état i, notée $P(j \to i)$.

Ah ! En fait je commence à comprendre pourquoi ce n'est pas possible d'en faire un exercice du BAC : la matrice est un peu trop grosse.

Et même si l'on ne tient pas compte de l'état 1, ce qui revient à considérer la matrice d'Alain Busser, une matrice carrée de taille 3 est certainement encore trop grosse.

ALGORITHME

Commençons par un trajet.

Le principe est exactement celui décrit dans le sujet : tant que Tom n'est pas dans l'eau ($y \geq -1$ et $y \leq 1$) et qu'il n'a pas atteint l'autre berge ($x \leq 9$), il avance.

```
p = "Tom";
t = 200;
Move(p,0,0); //on se place au centre du pont

while(Y(p)>=-1 && Y(p)<=1 && X(p)<=9){
    d=Math.ceil(Math.random()*3)-2;
    Pause(t);
    Move(p,X(p)+1,Y(p)+d);
}
```

Maintenant, si l'on veut estimer la probabilité de traverser avec succès, on encapsule tout cela dans une boucle et après chaque marche, on évalue si Tom est bien arrivé ou pas.

À noter qu'une traversée est réussie si Tom n'a pas glissé sur la berge au dernier moment…

```
p="Tom";
N=Input("Nombre d'essais ?");
compteur=0;
t=000;

for (i=1; i<=N; i=i+1){
    Move(p,0,0);
    while(Y(p)>=-1 && Y(p)<=1 && X(p)<=9){
        d=Math.ceil(Math.random()*3)-2;
        Pause(t);
        Move(p,X(p)+1,Y(p)+d);
    }

    if (X(p)==10 && Math.abs(Y(p))<=1){
        compteur=compteur+1;
    }
    SetText("Text1","Nombre d'essais : _i");
    SetText("Text2","Essais réussis : _compteur");
}
```

ÉTUDE THÉORIQUE

Il nous faut pour cela diagonaliser la matrice M et l'on obtient :

$$M = PDP^{-1} \qquad D = \begin{pmatrix} \frac{1-\sqrt{2}}{3} & 0 & 0 & 0 \\ 0 & \frac{1+\sqrt{2}}{3} & 0 & 0 \\ 0 & 0 & \frac{1}{3} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad P = \begin{pmatrix} 1 & 1 & 0 & 1 \\ -\frac{2+\sqrt{2}}{2} & \frac{\sqrt{2}-2}{2} & 1 & 0 \\ 1+\sqrt{2} & 1+\sqrt{2} & 0 & 0 \\ -\frac{2+\sqrt{2}}{2} & \frac{\sqrt{2}-2}{2} & -1 & 0 \end{pmatrix}$$

C'est vraiment pas beau… et là, on peut vraiment comprendre pourquoi il n'est pas possible d'en faire un exercice de BAC.

Je ne vais même pas écrire M^n : le fichier Maxima est joint.

NUMÉRIQUEMENT

Si l'on calcule $u^{(10)} = M^{10}\begin{pmatrix}0\\0\\1\\0\end{pmatrix}$, puisque l'on part de l'état ❸, on obtient bien $u^{(10)} \approx \begin{pmatrix}0.8625\\0.040272\\0.056953\\0.040272\end{pmatrix}$, comme dans le sujet.

On peut même calculer $u^{(4)} \approx \begin{pmatrix}0.4839\\ \cdots \\ \cdots \\ \cdots\end{pmatrix}$ et $u^{(5)} \approx \begin{pmatrix}0.5926\\ \cdots \\ \cdots \\ \cdots\end{pmatrix}$ qui nous montre que pour avoir au moins une chance sur deux de pouvoir traverser, le pont ne devrait pas dépasser pas les 4 pas… c'est assez peu !

Mais j'y pense… un robot… ça se téléporte, non ?

TÉLÉPORTATION

Faisons l’hypothèse supplémentaire que notre robot, suffisamment avancé technologiquement, possède la faculté de se téléporter.

Il n’est cependant pas fou :

- il sait que s’il est au milieu du pont, le prochain pas ne le fera jamais tomber à l’eau : inutile de se téléporter ;
- s’il est sur l’un des bords du pont, il possède une probabilité non négligeable (0.5) de tomber à l’eau. Dans ce cas-là, il a tout intérêt à se téléporter juste après avoir effectué son pas.

  Le problème est que l’eau faisant rouiller Tom instantanément, il doit donc prendre la décision de se téléporter avant d’effectuer son pas (sinon, ce serait trop facile).

Deux versions sont envisageables :

- Tom est capable de se téléporter n’importe où mais la quantité d’énergie nécessaire à la localisation de l’endroit précis ne lui permet de se téléporter que d’une case vers la gauche ou vers la droite ;
- Tom peut consacrer son énergie à sauter plusieurs cases, mais il ne peut alors se téléporter qu’en un seul endroit, décidé à l’avance et une fois pour toutes.

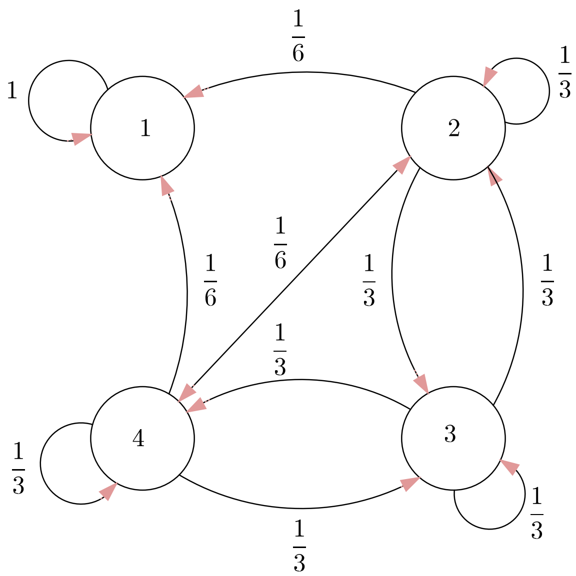
Version 1

Le système de téléportation de Tom lui permet de se téléporte d'une case vers la gauche ou vers la droite.

Ainsi, de l'état ❷, il pourra au final rejoindre tous les autres états : les mêmes qu'avant mais aussi l'état ❹ car après s'être déplacé en ❸, il peut se téléporter vers la droite, donc en ❹.

En partant de l'état ❷, Tom passe à l'état ❶ avec une probabilité de $\frac{1}{3}$ mais se téléporte en ❷ une fois sur deux donc $P(2 \to 1) = \frac{1}{6}$. Toujours en partant de l'état ❷, Tom reste dans le même état avec une probabilité de $\frac{1}{3}$ mais se téléporte en ❸ une fois sur deux donc $P(2 \to 2) = \frac{1}{6}$ à laquelle il ne faut pas oublier de rajouter $\frac{1}{6}$ provenant de la téléportation depuis ❶. Enfin $P(2 \to 3) = \frac{1}{6} + \frac{1}{6}$ et $P(2 \to 4) = \frac{1}{6}$.

D'où le graphe suivant :



Et l'algorithme :

```
p="Tom";
N=Input("Nombre d'essais ?");
compteur=0;

for (i=1; i<=N; i=i+1){
    Move(p,0,0);
    while(Y(p)>=-1 && Y(p)<=1 && X(p)<=9){
        if(Math.abs(Y(p))==1 && Math.random()>0.5){
            t = -Y(p)/Math.abs(Y(p)); //+ ou - 1
        } else {
            t = 0;
        }
        d=Math.ceil(Math.random()*3)-2;
        Move(p,X(p)+1,Y(p)+d+t);
    }

    if (X(p)==10 && Math.abs(Y(p))<=1){
        compteur=compteur+1;
    }
    SetText("Text1","Nombre d'essais : _i");
    SetText("Text2","Essais réussis : _compteur");
}
```

Quelques simulations nous donnent une probabilité de succès avoisinant les 36%.

Une rapide application numérique avec Maxima, où M représente la matrice associée au graphe nous donne :

```
(%i2) u:(M^^10).[0,0,1,0], numer;
      u[2]+u[3]+u[4], numer;
```

$$(\%o2) \quad \begin{bmatrix} 0.63262217872022 \\ 0.11530495991676 \\ 0.13676790144626 \\ 0.11530495991676 \end{bmatrix}$$

$$(\%o3) \quad [0.36737782127978]$$

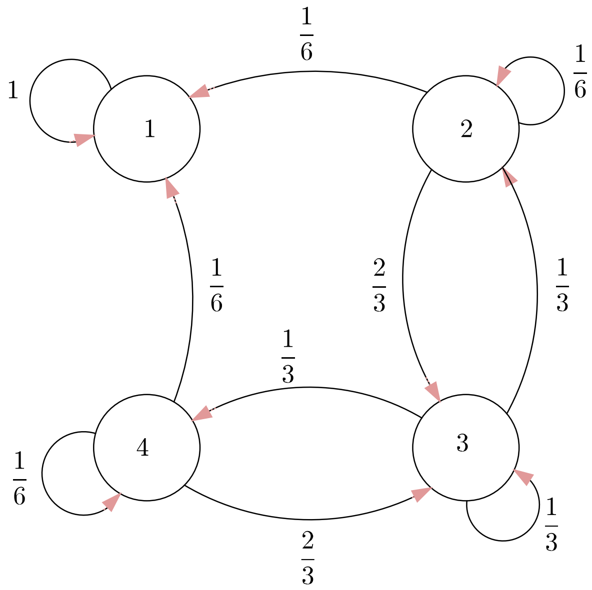
Pas fameux…

“YOU WILL BE UPGRADED”

Version 2

Tom se téléporte uniquement au centre du pont.

À partir de l'état ❷, on passe à l'état ❶ avec une probabilité de $\frac{1}{3}$ mais on se téléporte en ❸ une fois sur deux, donc $P(2 \to 1) = \frac{1}{6}$. De même pour $P(2 \to 2)$ et $P(2 \to 3)$ mais il faut rajouter à ce dernier $3 \times \frac{1}{6} = \frac{1}{2}$.



Algorithmiquement, cela se traduit par :

```
p="Tom";
N=Input("Nombre d'essais ?");
compteur=0;

for (i=1; i<=N; i=i+1){
   Move(p,0,0);
   while(Y(p)>=-1 && Y(p)<=1 && X(p)<=9){
      //si Tom est sur un bord, et avec une proba de 0.5
      if(Math.abs(Y(p))==1 && Math.random()>0.5){
         //il se téléporte au centre
         Move(p,X(p)+1,0);
      } else {
         d=Math.ceil(Math.random()*3)-2;
         Move(p,X(p)+1,Y(p)+d);
      }
   }

   if (X(p)==10 && Math.abs(Y(p))<=1){
      compteur=compteur+1;
   }
   SetText("Text1","Nombre d'essais : _i");
   SetText("Text2","Essais réussis : _compteur");
}
```

Quelques simulations nous montrent que la probabilité d'arriver sur l'autre berge avoisine maintenant les 50%, ce qui se vérifie numériquement en notant M la matrice associée au graphe :

```
(%i2) u0:[0,0,1,0]$
      u:(M^^10).u0, numer;
      u[2]+u[3]+u[4];
```

$$
(\%o3) \quad \begin{bmatrix} 0.53100794731918 \\ 0.10990656991439 \\ 0.24917891285204 \\ 0.10990656991439 \end{bmatrix}
$$

```
(%o4) [0.46899205268082]
```

C'est mieux…