

Notons que lorsque nous avançons en suivant une méthode déterminée, nous pouvons ne jamais revenir au point de départ et par exemple parcourir des arcs qui, à partir d'un certain moment, forment un cycle a-b-c-d-e-c-d-e-c-d-e-c-d-e... Les questions posées ne sont donc pas évidentes.

Solution: Quand vous suivez la méthode 1 et que vous êtes sur un arc dans une direction donnée, vous savez d'où vous venez: en vous retournant et en allant au nœud dont vous venez, l'arc qui vous y a amené est celui de gauche.

Il n'y a qu'un nombre fini de positions possibles sur un arc du graphe et seulement deux directions possibles à partir d'un nœud, donc il n'y a qu'un nombre fini de parcours d'un arc. Quand vous êtes sur un arc dans une direction donnée, la suite de ce que vous allez faire est déterminée. Il en résulte qu'en appliquant la méthode «toujours à droite», vous finirez par tomber sur un cycle. On est donc dans une situation du type a-b-c-a-b-c-a-b-c... ou du type a-b-c-d-e-c-d-e-c-d-e... Dans ces suites, les lettres minuscules désignent des positions sur un arc, donc un arc avec l'indication d'une direction de parcours.

Deux cas sont possibles: soit on entre tout de suite dans un cycle, soit on y entre après quelques parcours d'arcs où l'on ne repassera plus. Il faut montrer que le second cas est impossible. Si on avait une situation du type a-b-c-d-e-c-d-e-c-d-e..., cela signifierait qu'il y a deux antécédents à c: de l'arc orienté b on passe à c, et de l'arc orienté e on passe à c, cela en suivant la règle «toujours à droite». C'est impossible puisque, comme on l'a expliqué, on peut savoir d'où l'on vient: ou bien, quand on est en position c, on vient de la position b, ou bien quand on est en position c on vient de la position e. Un seul cas est possible. Puisque la situation où l'on entre dans un cycle après le parcours de quelques arcs par lesquels on ne repassera pas est impossible, cela signifie qu'on entre dès le départ dans un cycle, et donc qu'on revient à son arc de départ avec le même sens de déplacement sur l'arc.

Pour le second algorithme, le raisonnement est assez proche, sauf que cette fois on va considérer les parcours en prenant en compte la liste des arcs et sens de déplacement de l'arc par lesquels on passe, auxquels on ajoute l'information indiquant si l'on doit tourner à droite ou à gauche au prochain nœud. Un parcours sera donc la donnée d'une suite d'arcs orientés (désignés comme précédemment par des minuscules) avec pour chacun l'indication droite ou gauche. Par exemple a/droite—b/gauche—c/droite—d/gauche—...

Quand on est dans une position du parcours conçu de cette façon, on peut revenir en arrière, car l'information supplémentaire droite ou gauche nous indique aussi ce qui s'est passé au

nœud précédent et donc quel arc nous y a amenés. Comme dans le raisonnement précédent, il est clair qu'un parcours revient à un moment ou un autre en une position déjà occupée (car il n'y a qu'un nombre fini de positions x/droite y/gauche). Il faut prouver que cela ne peut pas se faire ailleurs qu'en la position de départ. Le raisonnement est le même que précédemment, avec la nouvelle notion de parcours: si on ne revenait pas au point de départ, l'une des positions aurait deux antécédents, ce qui n'est pas possible. Notons que le raisonnement assure qu'on revient à l'arc de départ en le parcourant dans le même sens et avec l'indication droite. Il n'est donc pas exclu qu'on y soit passé avant avec l'indication gauche ou en le parcourant dans l'autre sens, mais, quel que soit le cas, on revient à l'arc de départ.

Le résultat se généralise. Si, toujours avec un graphe dont tous les nœuds possèdent 3 arcs incidents, on applique une méthode de choix entre droite et gauche périodique, par exemple droite/droite/gauche répété, on sera certain de revenir à son point de départ. Pour le démontrer, on ajoute autant d'informations qu'il faut aux éléments de ce qu'on considérera comme des positions dans un parcours de façon à pouvoir revenir en arrière, et on raisonne comme précédemment en montrant qu'une position ne peut pas avoir deux antécédents.

Pour les graphes quelconques, on peut encore généraliser le résultat en procédant comme suit. On suppose que le plus grand nombre possible d'arcs incidents pour un nœud est N . On suppose qu'à chaque nœud, les arcs sont numérotés de 1 à M avec $M \leq N$. On peut d'ailleurs imaginer que cette numérotation est fixée par celui qui suit le parcours quand il arrive à un nœud pour la première fois. On définit la méthode de parcours en se donnant une suite périodique d'entiers entre 1 et N (par exemple 233142331423314...). On décide de suivre le chemin qui, à chaque nouvel arc rencontré, choisira l'arc correspondant à l'entier K de la suite périodique auquel on est arrivé, cela en respectant la convention suivante: on compte les chemins à partir de celui par lequel on est arrivé, éventuellement en tournant plusieurs fois si l'entier K est supérieur au nombre d'arcs incidents au nœud où l'on se trouve. Par exemple, si on arrive par l'arc 2, qu'il y a 5 arcs arrivant sur le nœud où l'on se trouve et que $K=7$, on repartira par le chemin $2+7=9$ qui est le même que l'arc 4, puisqu'on compte les K arcs en tournant: 3, 4, 5, 1, 2, 3, 4.

Une telle méthode, par un argument d'antécédent unique (qui sera applicable si l'on associe suffisamment d'entiers à chaque arc quand on définit la notion de position), vous ramènera toujours à votre arc de départ. De tels algorithmes de parcours engendrent des promenades complexes tout en préservant l'assurance de ne jamais se perdre. ■

BIBLIOGRAPHIE

ÉNIGME 1

P. Winkler, **Mathematical Mind-Benders**, chapitre « The adventures of ant Alice », A K Peters/CRC Press, 2007.

ÉNIGME 2

J.-P. Delahaye, **Le désordre total n'existe pas**, *Pour la Science* n° 376, pp. 86-91, février 2009.

R. Graham et al., **Ramsey Theory**, John Wiley & Sons, 1990.

S. Radziszowski, **Small Ramsey numbers**, 2017 : <http://bit.ly/2h26G2o>

ÉNIGME 4

J.-P. Delahaye, **Persuader de son savoir sans le transmettre**, *Pour la Science* n° 399, pp. 88-93, janvier 2011.

O. Goldreich, **Probabilistic proof systems : A primer**, *Foundations and Trends in Theoretical Computer Science*, vol. 3(1), pp. 1-91, 2008.