Thésée et le dédale magique

Année 2017 - 2018

Ont participé :

Mathieu BOULAN (1re ITEC)
Zoé CANIZARES (Terminale S-SVT)
Théophile LAUSEIG (Terminale S-SI)
Louis LOFFICIAL (Terminale S-SI)
Quentin MONNEREAU (Terminale S-SVT)

Encadrés par DELMAIRE Christine, professeur de mathématiques
DE TEYSSIERE Armelle, professeur de mathématiques

Établissements : Lycée de la Mer, Gujan-Mestras (33)
Lycée Notre-Dame, Bordeaux (33)

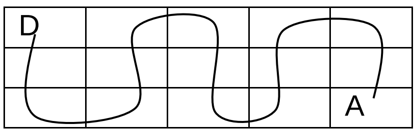
Chercheur : SOPENA Eric, chercheur au LaBRI à Talence (33)

Présentation du sujet

Eric Sopena nous a présenté ce sujet que nous avons appelé « Thésée et le dédale magique » : Un promeneur cherche à « visiter » toutes les cases d'un damier à m lignes et n colonnes, en respectant les règles de déplacement suivantes :

- La première case est toujours la case de coordonnées (1, 1), en haut à gauche.
- On passe d'une case à une autre horizontalement ou verticalement.
- On ne repasse jamais par une case déjà visitée.

Un exemple de promenade, sur un damier 3 x 5, est représenté ci-dessous :



On cherche alors à déterminer le nombre de promenades distinctes possibles sur un damier m x n, pour des valeurs de m et n fixées.

Nous démontrerons dans cet article une conjecture sur le nombre maximum de chemins possibles sur les grilles de taille 2 x n, et nous présenterons un algorithme qui calcule le nombre de promenades distinctes possibles sur un damier m x n.

Introduction

Avant de commencer nous allons tout d'abord expliquer les éléments de base du problème ; pour cela, et afin de mieux comprendre, nous avons décidé d'illustrer notre problème par les aventures menées par Thésée pour secourir Ariane.

Après sa cuisante défaite contre Thésée, le Minotaure décide de faire un pacte avec la déesse de la sagesse, Athéna et la déesse de la magie, Circée. Celui-ci dispose maintenant de terribles pouvoirs et décide de les utiliser pour créer un nouveau dédale magique pour piéger Thésée…

Le Minotaure a enfermé Ariane, la bienaimée de Thésée, dans le dédale, ce dernier n'a alors plus d'autre solution que de secourir la demoiselle en défiant le dédale magique.

Le minotaure utilise alors un sortilège pour téléporter Thésée, pendant son sommeil, dans une pièce rectangulaire…

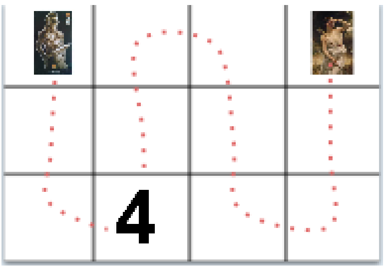
Il distingue alors au sol un carrelage, dont certains carreaux contiennent des nombres… Il en déduit qu'il est arrivé dans le dédale magique !

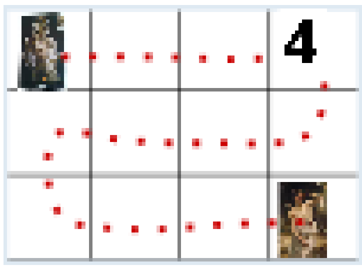
Soudain la voix du Minotaure retentit : « Si tu veux revoir ta belle Ariane, tu n'as point d'autre choix que de traverser le dédale ; marche sur tous les carreaux, sans passer deux fois sur le même. Il t'est impossible de te déplacer en diagonale. Ah, et aussi, pour finir, chacun de tes pas est numéroté, tu dois donc passer sur les bons carreaux au bon moment pour ne pas risquer de retourner au départ.

Bonne chance à toi, Ah ah ah !!! »

Ainsi en respectant les règles imposées par le Minotaure nous pouvons imaginer à quoi peut ressembler le trajet de Thésée.

On remarque alors que le choix de ces règles influe sur le parcours de Thésée, comme on peut le voir avec la numérotation des carreaux, car la position du 4 fait alors varier le chemin :





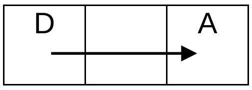
Thésée, est bien dans l'embarras…

Mais il connaît la formidable PUISSANCE des mathématiques, et décide de s'en servir pour trouver le bon chemin, avant de mourir de faim…

En se servant du fil d’Ariane, il va pouvoir établir le nombre de chemins possibles pour délivrer celle-ci, sachant que la difficulté réside dans la multitude des chemins possibles car le dédale bouge et les nombres de carreaux et de colonnes ou de lignes changent constamment…

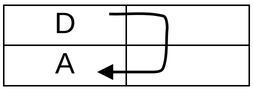
1. Les recherches :

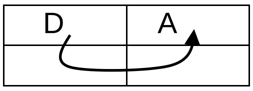
Nous avons commencé par des petites grilles et avons cherché tous les chemins possibles : (1)



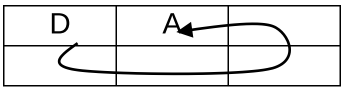
1 chemin possible

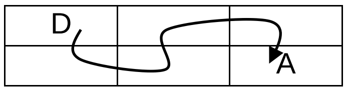
1 chemin possible





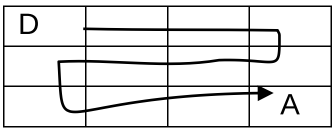
2 chemins possibles

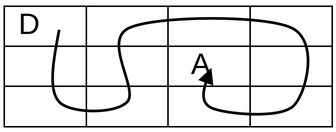


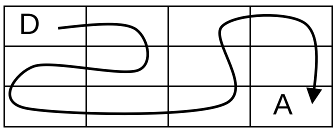


3 chemins possibles

Puis nous avons cherché sur des grilles plus grandes :







Mais très vite, nous avons vu qu’il n’était pas possible de dessiner tous les chemins sur des grilles plus grandes… Nous avons alors eu l’idée d’utiliser les arbres pour pouvoir tous les compter et, encore une fois, devant le nombre de plus en plus grand de possibilités, nous avons abandonné l’idée de tout faire à la main.
D’où l’idée de proposer un algorithme pour déterminer ce nombre.

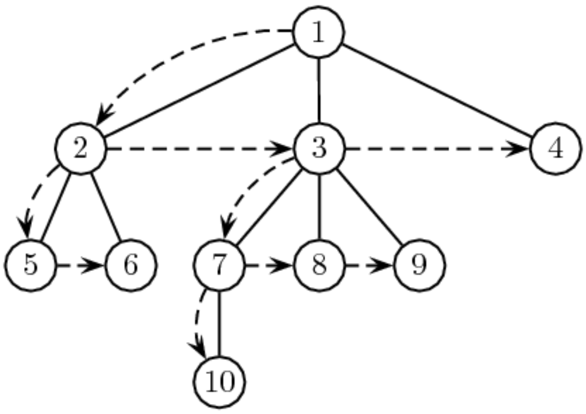
En effet, comme nous pouvons le voir, il existe une multitude de possibilités de chemins lorsque l’on part de la case départ ; par exemple, pour un quadrillage 6 par 3, on a déjà 78 solutions !

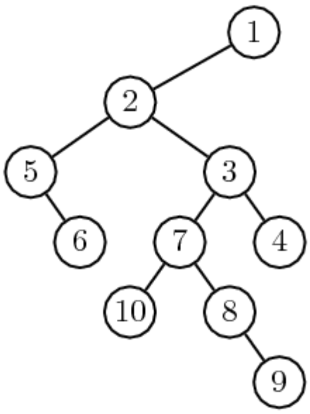
Ainsi nous verrons quelles stratégies Thésée peut mettre au point afin de déterminer le nombre de solutions…

2. Un puissant algorithme

Nous voyons bien que plus on augmente la taille de la grille, plus il y a de chemins. À titre d'exemple, pour une grille de 5 par 7 (pas si grande que ça), on a déjà 16 262 chemins !

Afin de contourner le problème, Thésée va utiliser l'incroyable puissance d'un **algorithme**. C'est une suite d'instructions que l'on peut traduire en langage informatique pour le faire exécuter par un ordinateur. En somme, c'est une sorte de recette de cuisine mais qu'on peut déléguer à une armée de cuisiniers ultra-performants, incroyable non ?
Donc Thésée va pouvoir déterminer les possibilités qui s'offrent à lui lors de son passage dans le labyrinthe, mais il lui manque une notion clef : la **récursivité**.
En informatique, la récursivité est, techniquement parlant, le fait qu'une fonction s'appelle elle-même, comme une sorte de suite. Notre héros en a bien besoin, en effet pour tester toutes les possibilités de chemins, il faut former un arbre de possibilités. Cela tombe bien car à chaque case 4 choix s'offrent à nous : aller en haut, en bas, à droite ou à gauche.
Donc la récursivité permet de faire ces choix en "appelant" les fonctions de déplacement à chaque case, tout en appelant le programme pour chaque case qu'il explore. Le concept est un peu obscur, voici une image pour mieux comprendre :
(2)





Le programme, comme sur le schéma, va se déplacer de case en case grâce au principe de récursivité. (3)

Maintenant que nous avons les connaissances nécessaires pour apréhender le programme, le voici en version simplifiée :

- Entrées
- Position initiale dans le tableau
- Appel récursif de la fonction (4 déplacements) /!\
  - Vérification des possibilités
  - Déplacement
- Génération des images (4)

L’avantage du programme est qu’il permet de tester les possibilités en simultané à partir du point de départ (comme sur le schéma), et donc de tester chaque possibilité, mais il a tout de même un assez gros inconvénient : il est gourmand en ressources.
En effet, quand les possibilités commencent à être trop nombreuses, l’ordinateur va devoir déterminer énormément de chemins et va donc mettre du temps avant de donner une réponse.

Cet aspect est normal, le problème donné ne peut être résolu que méthodiquement (c’est d’ailleurs ce que nous faisions avec de plus petits labyrinthes) et donc quand le dédale devient trop grand, même nos ordinateurs ont du mal.

Nous avons résumé les sorties de notre algorithme dans un tableau pour essayer de trouver une suite logique sur les différents résultats en fonction du nombre de lignes et/ou du nombre de colonnes, en vain… par faute de temps sans doute !

	1	2	3	4	5	6	7
1	1	1	1	1	1	1	1
2	1	2	3	4	5	6	7
3	1	3	8	17	38	78	164
4	1	4	17	52	160	469	1337
5	1	5	38	160	809	3478	16262
6	1	6	78	469	3478	22144	
7	1	7	164	1337	16262		

3. Un peu de mathématique :

On considère un quadrillage de L lignes et C colonnes. Ce quadrillage possède donc L × C cases. On attribue à chaque case le couple de coordonnées (L ; C). Thésée a été téléporté dans la case de coordonnées (1 ; 1). Combien de chemins Thésée peut-t-il parcourir dans ce quadrillage ? Soit F la fonction qui au couple (L ; C) associe le nombre de chemins possibles, que l’on note Y. on a donc :

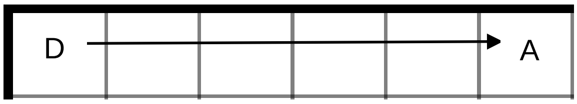
$$F : \mathbb{N}^2 \to \mathbb{N}$$
$$(L\ ;\ C) \mapsto Y$$

On remarque que $F(L\ ;\ C) = F(C\ ;\ L) = Y$.

Pour étudier cette fonction, on doit fixer l’un des paramètres et faire varier l’autre. On a choisit de fixer la variable L.

Étudions le cas où L = 1

On observe qu’on a toujours Y = 1, pour toutes les valeurs de C.



Cas où L = 1 et C = 6

Étudions le cas où L = 2

Propriété : Dans un quadrillage avec deux lignes et C colonnes, on a C chemins possibles, donc F(2 ; C) = C. De plus, il existe toujours un unique chemin dont la dernière case parcourue est une case de la dernière colonne.

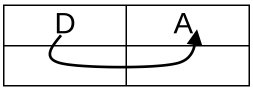
Démonstration :

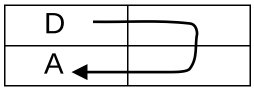
Utilisons le principe de récurrence.

Initialisation : Vérifions que la propriété est vraie pour C = 2

On part de la case (1 ; 1)

(1 ; 1) → (2 ; 1) → (2 ; 2) → (1 ; 2) ou (1 ; 1) → (1 ; 2) → (2 ; 2) → (2 ; 1)





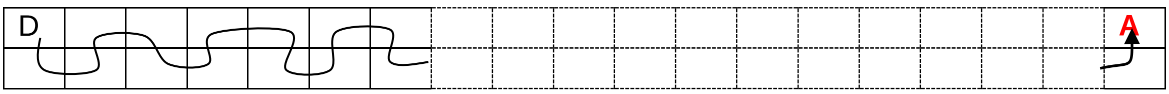
Soit F(2 ; 2) = 2, et il y a bien un unique chemin (le premier) qui se termine sur une case de la dernière colonne. La propriété est donc vraie pour n = 2.

Hérédité : Supposons qu'il existe un entier naturel n tel que F(2 ; n) = n, et qu'un seul chemin se termine sur une case de la dernière colonne. Montrons que la propriété est vraie au rang n + 1.

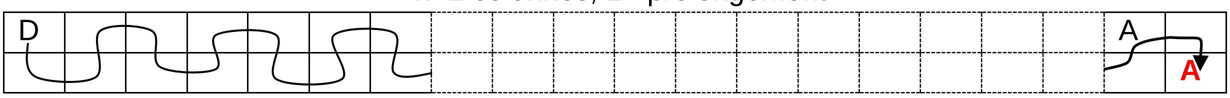
Dans un quadrillage de deux lignes et n colonnes, il y a n chemins possibles. De plus, parmi ces chemins, un seul d'entre eux a pour case finale une case de la forme (2 ; n) ou (1 ; n). On remarque que l'avant-dernière case parcourue est alors nécessairement l'autre case de la dernière colonne (car nous n'avons que deux lignes).

Donc, si l'on ajoute une colonne, on pourra prolonger ce chemin de deux façons possibles, et une seule de ces possibilités se terminera sur la dernière colonne :

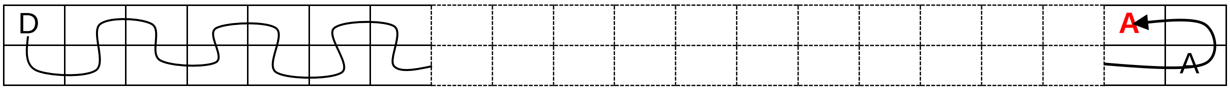
n colonnes



n+1 colonnes, 1er prolongement

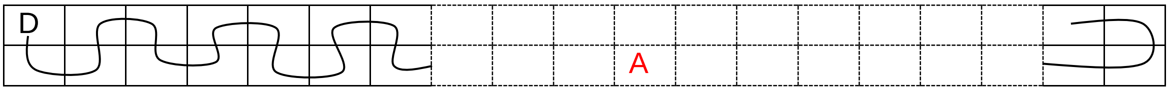


n+1 colonnes, 2e prolongement



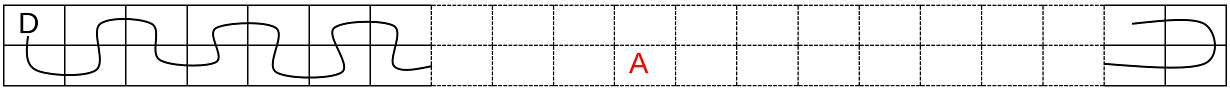
Les autres chemins sur le quadrillage de deux lignes et n colonnes ne se terminent pas sur la dernière colonne, et donc ils parcourent nécessairement ces deux cases en faisant un « aller-retour », de la façon suivante :

n colonnes, chemin ne se terminant pas sur la dernière colonne



Il n'y a alors qu'une seule façon de prolonger se chemin pour une solution à n+1 colonnes, et la case finale reste la même :

n+1 colonnes, chemin ne se terminant pas sur la dernière colonne



Nous avons donc bien au total n+1 chemins, et un seul d'entre eux se termine par une case de la dernière colonne.

Il nous reste à prouver qu'aucun autre chemin que ceux que nous avons construits n'est possible !

Considérons donc un chemin sur le quadrillage de deux lignes et n colonnes. S'il se termine sur une case de la dernière colonne, il n'y a alors que deux formes possibles de ce chemin sur les deux dernières colonnes (elles correspondent en fait aux deux solutions du cas n = 2), et ce sont donc bien les deux chemins que nous avons proposés dans un premier temps. Si ce chemin ne se termine pas sur la dernière colonne, alors il parcourt les deux dernières colonnes comme nous l'avons décrit plus haut. On peut alors le « raccourcir » (schéma inverse du schéma précédent) et retrouver l'un des chemins du cas à n colonnes…

Conclusion. La propriété est vraie pour n = 2, et est héréditaire. Donc, d'après le principe de récurrence, la propriété est vraie pour tout entier naturel n.

On notera que toutes les cases du quadrillage ne peuvent pas contenir des sorties, en effet, celles-ci sont réparties une fois sur deux, comme sur un damier.

4. Des recherches inabouties

Nous avons testé d'autres grilles en ajoutant une troisième ligne ou une quatrième ligne pour voir si on pouvait retrouver ce schéma pour les sorties :

3x3

D		A
	A	
A		A

3x4

D	A		A
A		A	
	A		A

3x5

D		A		A
	A		A	
A		A		A

…

4x4

D	A		A
A		A	
	A		A
A		A	

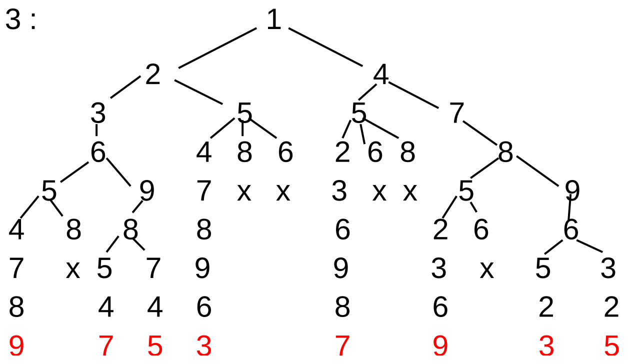
4x5

D	A		A	
A		A		A
	A		A	
A		A		A

Ces grilles semblent suivre cette règle, on le démontre en construisant des arbres.

Par exemple pour la grille 3 x 3 :

1	2	3
4	5	6
7	8	9



Conclusion

Thésée est dans une impasse et espère que le minotaure ne va pas multiplier les lignes et les colonnes de son dédale car sinon il passera sa vie entière à rechercher la belle Ariane !

En effet, nous savions qu’il y avait un très grand nombre de chemins possibles avec une progression exponentielle dès qu’on augmente le nombre de lignes et de colonnes grâce à notre algorithme, et nous n’avons pu déterminer le nombre maximum de chemins que pour les grilles de la forme 1 x m et 2 x m.

Nous avons pourtant cherché une logique dans la suite des nombres représentant les chemins possibles en fixant le nombre de lignes et en faisant varier le nombre de colonnes mais nous n’avons pas eu le temps de la trouver ! Alors, à vous de reprendre la quête de Thésée pour l’aider à sauver sa bien aimée…

Merci d’avoir lu notre article !

Notes d’édition

(1) Précisons que si la case départ est fixée, la case de sortie, elle, n’est pas fixée : on sort par n’importe laquelle des cases du bandeau extérieur.

(2) Ces schémas manquent d’explications :

Dans quel cas de damier travaille-t-on ?

Que signifient les nombres entourés de cercles ? Les flèches en pointillé ?

(3) Cette phrase est incompréhensible : ‘’ Le programme, comme sur le schéma, va se déplacer de case en case grâce au principe de récursivité.’’

(4) L’algorithme est trop esquissé et le code manque vraiment : les résultats annoncés dans le tableau de la page suivante perdent donc en crédibilité.