Fiche pédagogique

Activité : Crêpier Psychorigide

Objectifs pédagogiques : Apprendre à concevoir, expliciter et prouver un algorithme. Découvrir les notions de boucle et de complexité en temps.

Notions abordées : Algorithme, opérations non ambigues, variable, boucle, correction d'algorithme correct, complexité en temps.

Compétences développées :

⋆ **Chercher :**

- S'engager dans une démarche scientifique ;
- Organiser, tester et ajuster différentes stratégies pour résoudre un problème.

⋆ **Représenter :** Utiliser des instructions précises pour décrire un algorithme.

⋆ **Raisonner :**

- Concevoir un algorithme.
- Reconnaître des schémas, des configurations, des invariants.

⋆ **Communiquer :**

- Utiliser un vocabulaire précis et non ambigu pour décrire un algorithme.
- Expliquer sa démarche ou son raisonnement, comprendre les explications d'un autre et argumenter dans l'échange.

Matériel nécessaire : Des disques de tailles différentes, si possible numérotés par ordre de taille.

1. Ces disques peuvent être en papier, mais une certaine rigidité (en carton, en bois...) est préférable.
2. Le fait que les formes soient des disques n'est pas primordial, mais peut faciliter les choses. D'une part, dans un tas de disques superposés à peu près centrés, il est facile de voir si un disque est plus grand qu'un autre et il est immédiat d'identifier le plus grand ou le plus petit des disques. Cela est plus compliqué avec des carrés par exemple lorsque ceux-ci n'ont pas leurs côtés parallèles. D'autre part, avec des disques, il n'y a pas de perte de temps à l'opération parasite consistant réaligner les formes.

Niveau : A partir du cycle 3.

Durée : 30 minutes au minimum.

Sous licence **CC-BY-NC-SA** – Terra Numerica – 18-09-2026

Messages à faire passer

- Un algorithme est une séquence d'instructions simples pour effectuer une tâche complexe.
- Les instructions doivent être précises et non ambigües (pour pouvoir être comprises par un ordinateur). Par exemple, "Faire pareil", "le même truc", "celui-là" ne sont pas compréhensibles pas un ordinateur.
- Si une hypothèse (supposition) débouche sur une impossibilité, il faut l'éliminer. Se tromper n'est pas grave. Si on arrive pas à la solution, il est important de recommencer d'une autre manière.
- Il peut exister plusieurs algorithmes pour résoudre un même problème.
- Un algorithme doit être accompagné d'une preuve qu'il fonctionne comme annoncé.

Règle du jeu

Un apprenti cuisinier a préparé des crêpes (les disques), mais les a empilées n'importe comment. Son chef, psychorigide, veut les ranger de la plus grande (en bas) à la plus petite (en haut). Pour cela, un seul type de mouvement est possible : glisser une spatule entre deux crêpes pour retourner complètement le haut de la pile. Ici pour les participants cela revient à prendre le haut de la pile à partir d'une crêpe de son choix, de le soulever, le retourner et le reposer sur le bas de la pile. Voir Figure 1.Une telle opération est appelé un *retournement*. Le but du jeu est donc de faire des retournements successifs pour obtenir une pile bien rangée de la plus grande crêpe en bas à la plus petite en haut.

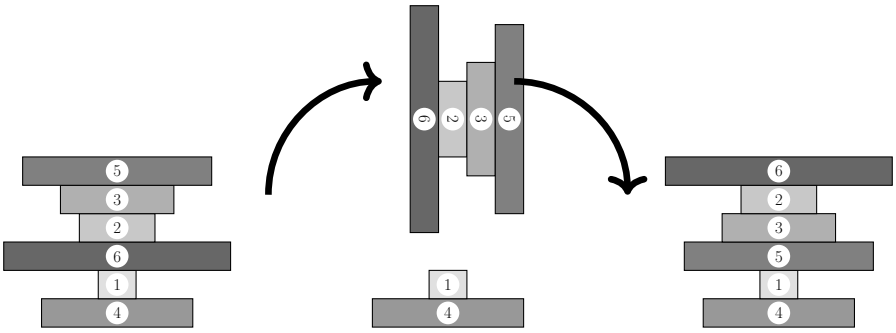


FIGURE 1 – Exemple de retournement. Les crêpes sont dans l'ordre 4-1-6-2-3-5, le haut de pile 6-2-3-5 est pris, retourné (il devient alors 5-3-2-6) et reposé sur le bas de pile 4-1. Les crêpes sont alors dans l'ordre 4-1-5-3-2-6.

Déroulement :

Présentation du jeu On met les participants par deux (mais ils peuvent être seul si l'on préfère) et on distribue à chaque groupe un jeu de 6 ou 7 disques de tailles différentes empilés de manière quelconque. On explique ensuite les règles du jeu. On explique qu'on doit enchainer les retournements et on montre différents retournements dont celui qui consiste à retourner la pile entièrement et celui qui consiste à ne retourner que la crêpe du haut.

Sous licence **CC-BY-NC-SA** – Terra Numerica – 18-09-2026

Conseil pédagogique

Il est pratique que les disques/crêpes soient numérotés. Le plus petit est numéroté 1, le deuxième plus petit numéroté 2, et ainsi de suite. Cela permet de référer facilement à une position. Cela a plusieurs avantages. Cela permet de nommer les différentes crêpes de manière rapide (il est plus facile de dire la crêpe 3 que la troisième plus petite crêpe) et surtout de les identifier rapidement (il est plus facile de repérer le numéro 3 sur la crêpe que de voir laquelle est la troisième plus petite sans numéro). Ensuite, cela permet d'identifier les différentes configurations en prenant l'ordre des crêpes à partir du bas. Voir par exemple légende de la figure 1. Il est préférable de prendre l'ordre de bas en haut plutôt que l'inverse, car quand on construit une pile, on commence par mettre la crêpe du bas. Le but du jeu est donc de mettre les crêpes dans la configuration 6-5-4-3-2-1.

Recherche Une fois le but et les règles du jeu bien comprises, on fait chercher aux participants un algorithme pour le résoudre. Pour cela on procède par étapes.

1. Tout d'abord, on fait une phase de découverte où on laisse les participants se familiariser avec le jeu et résoudre le problème sur une ou deux configurations.
2. Ensuite on leur demande de résoudre le problème à partir d'une nouvelle configuration afin de proposer un algorithme pour le proposer.
3. Enfin on leur demande de résoudre à nouveau le problème et de proposer un algorithme général qui peut le résoudre quelle que soit la configuration de départ.

Conseil pédagogique

Pour motiver à la recherche d'un algorithme générique, plutôt qu'une compilation de résolution particulière pour chaque configuration, on peut faire estimer aux participants le nombre de configurations avec 6 crêpes. Il y en a $6! = 6 \times 5 \times 4 \times 3 \times 2 \times 1$. On peut leur faire deviner de la manière suivante : demander combien il y a de postions possibles pour crêpe 6. Ils doivent normalement répondre 6. On demande ensuite combien il y a de positions possibles pour la crêpe 5, une fois que la crêpe 6 est placée. Il doivent répondre 5. Comme cela est vrai pour chacune des 6 positions possibles pour la crêpe 6, il y a donc 6×5 possibilités différentes pour les positions des crêpes 5 et 6. On demande alors combien il y a de positions possibles pour la crêpe 4, une fois que les crêpes 5 et 6 sont positionnées. Il doivent répondre 4. Et ainsi de suite jusqu'à arriver à $6 \times 5 \times 4 \times 3 \times 2 \times 1$.

En fonction de la manière dont les participants vont réfléchir, les trois étapes ne sont pas forcément clairement distinctes. Il se peut que certains aillent plus vite que prévu. Toutefois, cela se passe en général comme suit.

- Après la première étape, les participants ont tous réussi à mettre leur pile dans l'ordre, de manière plus ou moins contrôlée. Mais le plus souvent ils ne se rappellent pas ou ne savent pas expliquer ce qu'ils ont fait. On leur propose donc la deuxième étape en leur disant que le but est de trouver un algorithme pour résoudre le problème.
- A la fin de la deuxième étape, ils ont souvent trouvé une suite d'opérations pour résoudre le problème pour la configuration particulière qu'ils avaient à résoudre, mais le plus souvent ils ont du mal à l'expliciter de manière précise et leur procédure n'est pas générale. Il convient donc de leur parler des opérations qui peuvent être utilisées (voir paragraphe "Opérations compréhensibles" page 4). Si leur algorithme n'est pas générique, on leur montre une configuration sur laquelle il ne marche pas.

Sous licence **CC-BY-NC-SA** – Terra Numerica – 18-09-2026

– A la fin de la troisième étape, les participants ont généralement trouvé l'algorithme, mais de manière parfois partielle ou trop peu précise. On explique alors un algorithme construit avec eux, on le détaille et on demande de prouver sa validité. Voir paragraphe "Un algorithme possible" page 4.

Conseil pédagogique

Si beaucoup de groupes ont du mal à trouver une solution, on peut proposer de faire la résolution en commun à partir de la même configuration de départ. (Dans ce cas, il est très pratique d'avoir les crêpes numérotées.) Chacun leur tour, les groupes proposent un retournement que tout le monde effectue. En général, cela donne lieu à débat car certains peuvent ne pas être d'accord avec ce qui est proposé. En cas de désaccord, demander à chacun d'expliquer son choix de retournement. Cela aide normalement à faire émerger collectivement un algorithme.

Conseil pédagogique

Comme d'habitude, tout au long de l'atelier, il est important d'énoncer les messages dans l'encart précédent. Il est également bénéfique de prendre quelques minutes à la fin de l'atelier pour les rappeler.
En fin d'atelier (et seulement à la fin), il peut être utile de leur demander ce qu'il ont fait et de leur dire ou faire dire que c'est de l'informatique. Ceci permet d'ancrer l'idée que l'informatique peut être ludique et ne se fait pas nécessairement face à un écran.

Didactique et pédagogie

Opérations compréhensibles Une des principales difficultés des néophytes en algorithmique n'est pas d'imaginer un algorithme mais de l'exprimer clairement d'une part et ensuite de manière suffisamment précise et non-ambigue pour qu'il soit "compréhensible" par un ordinateur.

Une tendance naturelle est de montrer une crêpe avec le doigt. Une règle simple pour éviter cet écueil et d'interdire de montrer et de forcer à désigner une crêpe uniquement avec des mots. Il y a deux manières d'identifier une crêpe.

– Par sa taille ou son numéro, avec des descriptions du type "la troisième plus grande crêpe", "la quatrième plus grande crêpe", ou "la crêpe numéro 4". (Dans le cas de 6 crêpes, ces trois formulations désignent la même crêpe.)
– Par sa position dans la pile, avec des descriptions comme "la troisième crêpe en partant du haut", "la quatrième crêpe en partant du bas". (Là aussi, avec 6 crêpes, ces deux formulations désignent la même crêpe.)

Une fois bien clarifié ce qu'est une opération "compréhensible" par un ordinateur, on peut restreindre les participants à ne désigner les crêpes que par les deux moyens susdécrits.

Une deuxième chose classique est le fameux " On fait la même chose". En fait, quand quelqu'un dit cela, il ne veut en général pas que la même chose stricto sensu soit effectuée. En fait, la personne veut faire une opération similaire avec un paramètre qui a changé. Très souvent, ce changement de paramètre est tellement évident qu'elle ne s'en rend pas compte. Il est donc important de faire en sorte qu'elle s'en rende compte. Typiquement, on peut faire le robot idiot qui applique l'algorithme. Quand on arrive à une instruction du type "On fait la même chose" alors on fait littéralement la même chose (c'est-à-dire sans le changement de paramètre) afin que la personne s'aperçoive qu'on ne fait pas ce qu'elle souhaite, proteste et explique ce qui a changé.

Sous licence **CC-BY-NC-SA** – Terra Numerica – 18-09-2026

De manière générale, quand un participant énonce une instruction ambigue, on peut faire le robot idiot et l'appliquer délibérément d'une manière qui n'est pas celle que le participant souhaite pour le forcer à préciser.

Un algorithme possible

Il est important de ne pas imposer l'algorithme qui résoudra le problème. La première chose qui prévaut est la validité de l'algorithme. Tout algorithme valide est recevable. Toutefois, les participants convergent en général vers un algorithme tel que celui que nous allons décrire. Et on peut les y guider pour y arriver.

L'idée principale qui permet de résoudre le problème est la suivante : On peut ranger la plus grande crêpe dans la position voulue (en bas) en la mettant d'abord en haut puis en retournant toute la pile. Puis on recommence en ignorant les crêpes déjà rangées. Cela est correct et typique d'un algorithme récursif, mais demande de savoir bien spécifier quelles sont les crêpes rangées.

Il nous parait plus simple et instructif de les guider vers un algorithme itératif. Pour cela, on peut leur demander d'expliciter clairement les différentes étapes de l'algorithme, d'abord la première, puis le seconde, etc... On arrive alors à un algorithme qui ressemble à celui ci-dessous.

Etape 1 (qui permet de bien positionner la plus grande crêpe (*crêpe 6*) :

a. Retourner le haut de pile à partir de la plus grande crêpe (*crêpe 6*).
b. Retourner toute la pile (*le haut de pile à partir de la position 1*).

Etape 2 (qui permet de bien positionner la crêpe 5) :

a. Retourner le haut de pile à partir de la crêpe 5.
b. Retourner le haut de pile à partir de la position 2.

Etape 3 (qui permet de bien positionner la crêpe 4) :

a. Retourner le haut de pile à partir de la crêpe 4.
b. Retourner le haut de pile à partir de la position 3.

Etape 4 (qui permet de bien positionner la crêpe 3) :

a. Retourner le haut de pile à partir de la crêpe 3.
b. Retourner le haut de pile à partir de la position 4.

Etape 5 (qui permet de bien positionner la crêpe 2) :

a. Retourner le haut de pile à partir de la crêpe 2.
b. Retourner le haut de pile à partir de la position 5.

Une fois l'étape 5 écrite, on peut soit leur faire écrire l'étape 6, soit leur faire directement remarquer qu'une fois que les crêpes 6, 5, 4, 3, 2 sont à la bonne place alors la crêpe 1 l'est aussi.

Une fois qu'on a un algorithme tel que le précédent, on leur faire remarquer les invariants. Dans le cas présent, qu'à chaque étape i, on retourne le haut de pile à partir de la crêpe $7-i$, puis on retourne la pile à partir de la position i. Suivant le niveau, on montre comment ou on demande de factoriser l'algorithme dans la boucle suivante :

Pour i **allant de** 1 **à** 5, **faire :**

a. Retourner le haut de pile à partir de la crêpe $7-i$.
b. Retourner le haut de pile à partir de la position i.

A partir de cet algorithme, on leur demande comment généraliser à une pile à n crêpes. Il y a deux choses à modifier, qui dépendaient effectivement du nombre de crêpes : le nombre de boucles à faire (passer de 5 à $n-1$) et le numéro de la $i$ème plus grande crêpe (passer de $7-i$ à $n+1-1$). On obtient donc l'algorithme suivant :

Sous licence **CC-BY-NC-SA** – Terra Numerica – 18-09-2026

Pour i **allant de** 1 **à** $n-1$**, faire :**

a. Retourner le haut de pile à partir de la crêpe $n+1-i$.

b. Retourner le haut de pile à partir de la position i.

Une fois l'algorithme décrit, on peut leur faire appliquer cet algorithme sur un exemple. Il est important de montrer que certaines étapes sont inutiles, mais que l'algorithme les effectue quand même. Par exemple, si on part de la configuration 3-2-6-4-1-5, l'algorithme se déroule comme décrit Figure 2. On peut remarquer que l'étape 3.a. ne sert à rien car on retourne uniquement la crêpe du haut. De même la pile de crêpes est triée après l'étape 3.b. Cependant, l'algorithme continue de s'exécuter jusqu'à la fin.

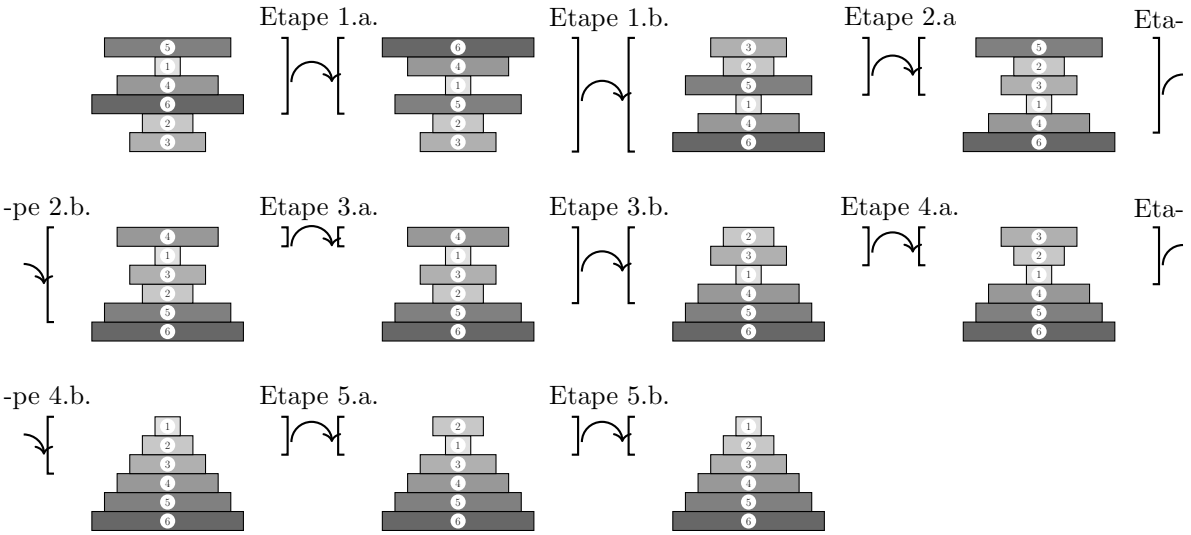


FIGURE 2 – Exemple de déroulement de l'algorithme de tri des crêpes

Validité de l'algorithme Une fois un algorithme clairement explicité, on demande aux participants de prouver (ou au moins donner des arguments) la validité de l'algorithme. On insiste sur le fait qu'un algorithme doit être accompagné d'une preuve.

Ici la preuve est assez simple. Elle repose sur fait qu'à chaque étape i on range la crêpe $n-i$. Cela se montre simplement par récurrence. Au début de l'étape i, toutes les plus grandes crêpes jusqu'au numéro $n+2-i$ ont été rangées. La crêpe $n+1-i$ est donc strictement au dessus de la position $i-1$. En effectuant l'opération i.a. on donc mettre la crêpe $n+1-i$ en haut de la pile sans toucher les crêpes déjà rangées. Ensuite en faisant l'opération i.a., la crêpe $n+1-i$ va se retouver à la position i qui est celle où elle doit être rangée. On a donc alors toutes les plus grandes crêpes jusqu'au numéro $n+1-i$ qui sont rangées.

Conseil pédagogique

A partir de la 4ème, on peut présenter cette récurrence de manière formelle. Pour les plus jeunes, on fait la preuve pour l'étape 1, puis l'étape 2 et on termine par un et ainsi de suite.

Complexité en temps On aborde ensuite la complexité en temps en faisant calculer le nombre de retournements effectués par l'algorithme. Comme on fait $n-1$ étapes de la boucle et deux retournements par étapes, l'algorithme fait en tout $2(n-1) = 2n-2$ retournements.

Sous licence **CC-BY-NC-SA** – Terra Numerica – 18-09-2026

A partir de la troisième, il est intéressant de faire cette activité avec les Tour de Hanoî. Cela permet de montrer la différence entre un algorithme rapide comme celui pour les crêpes, qui effectue un nombre linéaire d'opérations, et un algorithle lent comme celui pour les tours de Hanoï, qui effectue un nombre exponentiel (en l'occurence 2^{n-1}) d'opérations. Ainsi si on a 21 disques, l'algorithme pour les crêpes effectuera 40 opérations, alors que celui des tours de Hanoï en fera 1 048 576. A raison, d'une opérations par seconde, l'algorithme des crêpes s'effectue en 40 secondes, alors celui des tours de Hanoï en plus de 12 jours et 3 heures.

Autres références :

- `https://irem.univ-grenoble-alpes.fr/medias/fichier/fiche-prof-crepier-psychorigide_1558357226034-pdf?ID_FICHE=52380&INLINE=FALSE`
- `https://interstices.info/genese-dun-algorithme/`
- `https://atelier-canope-19.canoprof.fr/eleve/Formation%20initiale%20et%20continue/Robotique/Activites_debranchees/activities/Activites_debranchees_5.xhtml`
- W. H. Gates, C. H. Papadimitriou : Bounds for sorting by prefix reversal. Discret. Math. 27(1) : 47-57 (1979)
- M. Heydari and I. Sudborough. On the diameter of the pancake network. Journal of Algorithms, 25(1) :67-94, October 1997
- B. Chitturi, W. Fahle, Z. Meng, L. Morales, C.O. Shields, I. Sudborough, and W. Voit. An (18/11)n upper bound for sorting by prefix reversals. Theoretical Computer Science, 410(36) :3372-3390, 2009.
- L. Bulteau, G. Fertin, I. Rusu : Pancake Flipping is hard. J. Comput. Syst. Sci. 81(8) : 1556-1574 (2015) `https://arxiv.org/abs/1111.0434`
- G. Jäger, N. Oijid : Exact number of flips required to sort a burnt stack of pancakes. CoRR abs/2601.09447 (2026) `https://arxiv.org/abs/2601.09447`

Sous licence **CC-BY-NC-SA** – Terra Numerica – 18-09-2026