Interventions d'informatique débranchée en classe

Marmottes aux sommeils léger

1 Les marmottes au sommeil léger

L'objectif de cette activité est de comprendre, par expérimentations, l'utilité de la **compression de texte** et de l'appliquer sur un exemple simple (compression, transmission, décompression). Elle permet également d'aborder brièvement la notion d'**arbre binaire**.

L'activité est calibrée pour une séquence de 45 minutes, avec des élèves de cycle 3. Ces derniers sont regroupés par groupe de 5 ou 6, dans l'idée de promouvoir la collaboration au sein du groupe et l'émergence d'un design optimal pour le terrier. En fait, on aimerait que les élèves s'approchent d'une solution optimale par petites améliorations – échange de marmottes, déplacement d'une branche de terrier – par remarques successives au sein du groupe.

Préparation en amont

Télécharger l'annexe, découper les embranchements de terrier et les marmottes. Sur les marmottes, noter ces couples de nombre-lettres (chiffre sur la marmotte, lettre *à l'envers* au dos) suivants :

$$(N,6),(S,7),(I,7),(C,3),(O,5),(M,3),(P,2),(L,5),(E,12)$$

Sur les embranchements, de dos et coude vers le bas, noter 0 sur la branche gauche et 1 sur la droite. Ensuite, plastifier pour que le jeu vive plus longtemps.

Remarque : Il est possible de plastifier avant de noter les couples nombre-lettre et d'utiliser un feutre non permanent par dessus pour pouvoir jouer différentes instances du problème.

Mise en contexte du problème

L'hiver approche. Il est grand temps pour ce groupe de marmottes de construire un terrier pour hiberner tout l'hiver et être en pleine forme au printemps : aux élèves de les aider !

Un terrier se compose d'une **entrée**, par laquelle les marmottes entrent et sortent ; d'**embranchements de galeries**, qui permettent d'accéder à deux nouvelles galeries plus profondes (seulement deux pour éviter que le terrier ne s'effondre) ; et de **chambres**, situées à l'extrémité des galeries et ne pouvant accueillir qu'une seule marmotte (elles sont toutes petites mais en nombre suffisant pour que personne ne dorme dans les galeries et ne bloque le passage).

Mais il y a un problème : ces marmottes ont le sommeil léger et vont se réveiller plusieurs fois au cours de l'hiver – ce nombre est écrit à côté de chaque marmotte – et vont en profiter pour prendre l'air.

Pour que chacun dorme le mieux possible, la mission des élèves est de minimiser les déplacements au sein du terrier.

Exemple : Si une marmotte se réveille 4 fois pendant l'hiver et qu'elle a besoin d'emprunter 3 galeries pour remonter jusque la surface, elle a effectué $3 \times 4 = 12$ déplacements (on ne compte pas les déplacements du retour car la marmotte se laisse glisser jusque son lit, et donc ne se fatigue pas).

Objets de l'informatique

→ **Algorithme.** L'élève est amené à expérimenter avec le matériel pour résoudre le problème du codage de Huffman. Certains groupes peuvent parvenir à trouver des arbres optimaux et à avoir l'intuition de l'algorithme qui se cache derrière. Dans tous les cas, l'algorithme est affiché au tableau et les élèves auront alors la tâche de l'appliquer. L'intervenant est amené à la fin de l'activité à définir la notion, et à expliquer les questions de complexité et terminaison des algorithmes. On introduit également le concept d'arbre binaire et quelques éléments de terminologie autour de cela.

→ **Information.** La notion cachée derrière l'activité est celle de la compression de donnée par codage de Huffman. Il est également nécessaire d'expliquer la notion de codage.

→ **Langage.** Les élèves sont introduits au concept de codage. Le codage de Huffman est un code préfixe : aucune lettre n'a dans son codage en préfixe le codage d'une autre lettre. Cela permet d'assurer que le décodage d'un caractère est non-ambigue. Cette propriété peut être expliquée aux élèves en guise d'extension (très peu probable).

→ **Machine.** Pas applicable.

Recherche par groupe (15min)

Creuser son premier terrier

Dans cette première étape, il faut les **laisser expérimenter** tout en vérifiant bien que les règles de construction des terriers sont bien respectées. Veiller à bien mettre toutes les marmottes – on a besoin d'un couloir de moins que le nombre de marmottes – et prendre les papiers du bon côté (avec les nombres).

L'importance du sommeil léger

Encore une fois, il s'agit de les **laisser expérimenter** tout en vérifiant qu'il n'y a pas de marmottes aux croisements et que toutes les règles sont respectées. Vérifier que le calcul du **bruit** des terriers est maîtrisé.

Mise en commun

À cette étape, il s'agit, en plus de trouver le terrier le moins bruyant parmi les groupes, de réfléchir ensemble à une stratégie pour trouver le terrier le moins bruyant possible.

L'argument *celles qui se réveillent plus souvent doivent être proches de l'entrée*. Cela nous explique comment optimiser un terrier déjà construit mais pas comment le construire de manière optimale : tous les couloirs de la même longueur ? un peigne ? En fait, ça dépend du nombre de marmottes.

Explication de l'algorithme de Huffman (5min)

1. Détacher tout
2. Relier ensemble les marmottes qui se réveillent le moins souvent. En cas d'égalité, en choisir une.
3. Noter sur le coude du couloir la somme du poids des marmottes. Ce couloir sera maintenant considéré comme une seule grosse marmotte (difficile pour les enfants).
4. Reprendre à l'étape 2 jusqu'à ce qu'un seul terrier – contenant toute les marmottes – soit construit.

Laisser l'algorithme projeté au tableau pour que les enfants y accèdent facilement.

Construction du terrier optimal (10min)

On les laisse utiliser l'algorithme, calculer le poids du nouveau terrier, constater qu'il est plus faible que le terrier construit par tâtonnement.

Bien insister sur le fait que cela marcherait **quelque soit** le nombre de marmottes.

Pourquoi c'est de l'informatique ? (5min)

→ **Structure d'arbre.** C'est une structure de données, c'est-à-dire une façon de ranger des données. Dans cette structure, on utilise le vocabulaire des arbres, et plus particulièrement des arbres généalogiques : le noeud sans parent est la **racine**, ceux sans enfants sont les **feuilles**, deux noeuds qui ont le même parent sont dit **frères** (ou soeur). Il y en a de toutes les formes, certains avec toutes les branches de même taille, certains avec une forme de peignes etc. Dans notre cas, l'arbre est **binaire** car chaque noeud parent a deux enfants.

→ **Algorithme.** C'est un ensemble d'étapes à appliquer dans un certain ordre pour résoudre un problème donné. C'est un peu comme une recette de cuisine. C'est ce que vous avez utilisé pour construire le terrier optimal. En informatique, on se pose plusieurs questions sur les algorithmes : leur **terminaison**, c'est-à-dire est-ce qu'il y a un moment où on s'arrête ?, leur **vérification**, c'est-à-dire vérifier qu'ils font bien ce qu'on veut et leur **complexité**, c'est-à-dire le temps qu'ils mettent pour résoudre le problème donné.

→ **Compression.** C'est un procédé qui permet de réduire la taille des données volumineuses. Elles sont alors plus faciles à **stocker** sur un disque dur, ou à **envoyer** sur internet. On utilise la compression tout les jours sans s'en rendre compte, par exemple lorsque l'on regarde une vidéo. Dans certains cas, il faut décompresser les données avant de pouvoir à nouveau les utiliser. Il existe de nombreux **algorithmes** de compression, certains sont dits **sans perte** : on ne perd pas d'informations entre l'original et les données décompressées. Le codage de Huffman est un algorithme de compression sans perte, et est toujours utilisé de nos jours malgré son ancienneté.

Application au décodage d'un message (10min)

Trace écrite

L'hiver approche. Il est grand temps pour ce groupe de marmottes de construire un terrier pour hiberner tout l'hiver et être en pleine forme au printemps : il faut pour cela que les marmottes qui se réveillent le plus souvent soient proches de la sortie pour ne pas réveiller les autres.

C'est de l'informatique parce que nous avons utilisé un a_____ (un ensemble d'instructions) pour résoudre un problème sur une s_____ de donnée appelée a_____. Cette technique, découverte par Huffman, est encore utilisée aujourd'hui pour faire de la c_____ (utiliser le moins de 0 et de 1 possible pour décrire l'information) de fichiers `.mp3`, `.mp4`, `.zip`...