FENG David - A2 - 10250
KETTANI Kenza - A2- 10279

Projet Laboratoire

Rapport:
Tours de Hanoi magnétiques

Principe

Les tours magnétiques de Hanoi sont une variation des tours de Hanoi classiques où chaque disque possède deux faces de couleurs différentes: une face bleu (B) et une rouge (R). Le principe de ce jeu mathématique est le même que celui des tours classiques, mais avec deux contraintes supplémentaires:

- ❖ Chaque disque se retourne quand il est déplacé;
- ❖ Deux disques ne peuvent pas être placés l'un sur l'autre si les faces qui se touchent sont de la même couleur: pas de R/R ni de B/B;

Ce jeu est composé de trois piles A, B et C: A est la source (S), B la destination (D) et C la pile temporaire (I comme Intermédiaire). Au départ, la pile A contient un nombre N de disques empilés par ordre de taille: les disques sont empilés du plus large au plus petit, la face supérieure étant rouge. L'objectif du jeu est de déplacer toute la colonne, un disque à la fois, vers la pile B (en utilisant la pile intermédiaire I), tout en maintenant l'ordre original d'empilement des disques et que les faces bleues soient les faces supérieures.

En effet, les bases de chaque pile possèdent leurs propres couleurs. Nous avons la couleur Rouge/Bleu pour la pile A et la couleur Bleu/Rouge pour les piles B et C.

Ainsi, la fonction “mouvement” de cet algorithme est réglementée par trois conditions, considérées comme les erreurs de déplacement dans notre code:

1. Erreur de **taille**: un disque ne peut pas être déplacé sur un disque plus petit;
2. Erreur de **couleur**: deux faces de la même couleur ne peuvent pas être superposées;
3. Erreur de **déplacement**: mouvement impossible de B à C ou de C à B à cause des deux raisons précédentes.

Au total, nous avons identifié quatre déplacements possibles:

A ⇒ B

A ⇒ C

B ⇒ A

C ⇒ A

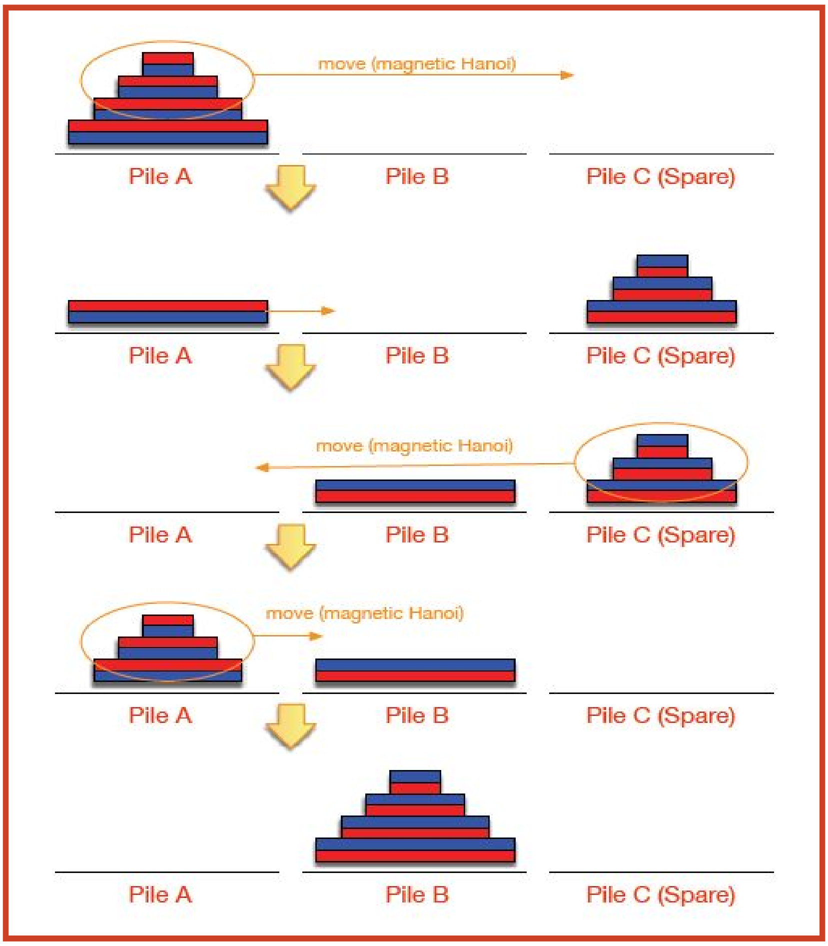
Concernant les déplacements **B ⇒ C** et **C ⇒ B**, il faut que le disque passe par la pile A avant que cette dernière se déplace vers C ou B.

Donc, en plus des quatre déplacements possibles, il faut considérer les deux déplacements supplémentaires :

B ⇒ A ⇒ C

C ⇒ A ⇒ B

Voici un schéma récapitulatif du déroulement de l’algorithme (toutes les étapes n’ont pas été représentées, par souci de clarification):


move (magnetic Hanoi)
Pile A
Pile B
Pile C (Spare)
Pile A
Pile B
Pile C (Spare)
move (magnetic Hanoi)
Pile A
Pile B
Pile C (Spare)
move (magnetic Hanoi)
Pile A
Pile B
Pile C (Spare)
Pile A
Pile B
Pile C (Spare)

Solution pour les Tour de Hanoï (classique)

Voici le C code permettant de résoudre les Tour de Hanoï pour la version classique:

```
hanoi (n, src, dst) {
        hanoi (n - 1, src, tmp);     /* tmp = pile temporaire C*/
        move (1, src, dst);
        hanoi (n - 1, tmp, dst);
}
```

Pour les déplacements, il suffit de vérifier qu'il n'y ait pas un problème de **taille**. Ainsi, nous avons 6 déplacements possibles pour ce cas:

A ⇒ B
A ⇒ C
B ⇒ A
B ⇒ C
C ⇒ A
C ⇒ B

Ainsi, pour définir les déplacements, nous procédons comme suit:

- If r5=0, r6=1, move A to B
- If r5=0, r6=2, move A to C
- If r5=1, r6=0, move B to A
- If r5=1, r6=2, move B to C
- If r5=2, r6=0, move C to A
- If r5=2, r6=1, move C to B

Solution pour les Tour de Hanoï Magnétiques

Voici le code C résumant les grandes étapes **récursives** décrites précédemment:

```
/* bouger les N disques de la pile source à la pile destination */
mhanoi (n, src, dst) {
        mhanoi (n - 1, src, tmp);  /* tmp = pile temporaire C*/
        move (1, src, dst);
        mhanoi2 (n - 1, tmp, src);
        mhanoi (n - 1, src, dst);
}

mhanoi2 (n, src, dst) {
        mhanoi2 (n - 1, src, dst);  /* tmp = pile temporaire C*/
        mhanoi (n - 1, dst, tmp);
        move (1, src, dst);
        mhanoi2 (n - 1, tmp, dst);
}
```

Pour résoudre le problème des déplacements B vers C ou bien C vers B, il faut implémenter un deuxième sous programme qu'on appelle ici **mhanoi2**.

Pour tout déplacement de disque, il faut vérifier que nous n'ayons pas de problème de **taille**, ni de **couleur**.

Ainsi, pour définir les déplacements, nous procédons comme suit:

- If r5=0, r6=1, move A to B
- If r5=0, r6=2, move A to C
- If r5=1, r6=0, move B to A
- If r5=1, r6=2, move B to C => erreur de couleur
- If r5=2, r6=0, move C to A
- If r5=2, r6=1, move C to B => erreur de couleur

Pour conclure, en traduisant correctement ces codes C en langage assembleur et en allouant correctement les mémoires à chaque étape, nous parvenons à résoudre ces jeux mathématiques, que ce soit pour les Tours de Hanoï version classique ou bien sa version dérivée les Tours de Hanoï Magnétiques.