L3 Informatique : Méthodes formelles d'aide à la détection d'erreur
Problème du robot : 2ème partie

DÉMONSTRATIONS

Remarque : On choisit la postcondition 2 (cf. Problème du robot : 1ère partie) où il peut y avoir moins de trois couleurs dans la donnée.

1. Démonstration de l'arrêt

Arrêt des instructions élémentaires et des appels de fonction.

Les instructions des lignes 1, 2, 4, 7, 10, 12 et 15 (voir numérotation des lignes ci-dessous) sont des instructions élémentaires (calculs d'expressions simples de types compatibles pour les comparaisons ou affectations dans des variables de types compatibles).

Les instructions des lignes 3, 6, 8 et 12 comportant des appels à fonction, il faut vérifier que les préconditions de ces fonctions sont bien respectées. L'exécution de la ligne 8 (respectivement la ligne 13) est conditionnée par le test de la ligne 7 (respectivement de la ligne 12), il reste donc uniquement à vérifier que iW, iR et iB sont bien des indices du tableau, lorsqu'ils sont paramètres de LireCouleur( ) ou Echanger( ).

```
Variables
  entier iB, iW, iR ;
Début
 1      iB ← 1 ; iW ← 1 ; iR ← N ;
        // P0 : 1 ≤ iB, iB ≤ iW, iR ≤ N
 2      Tant Que (iW ≤ iR) Faire
        // P1 : 1 ≤ iB ≤ iW ≤ iR ≤ N
 3          Si (LireCouleur(iW) = W) Alors
            // P1
 4              iW ← iW + 1 ;
                // P2 : (LireCouleur(iW) = W)_avant → ((1 ≤ iB < iW) ∧ (iR ≤ N))
 5          Sinon
            // P1
 6              Si (LireCouleur(iW) = R) Alors
 7                  SI (iW ≠ iR) Alors
                    // P3 : P1, LireCouleur(iW) = R, iW ≠ iR
 8                      Echanger(iW, iR) ;
 9                  FinSi
10                  iR ← iR – 1 ;
                    // P4 : (LireCouleur(iW) = R)_avant → ((1 ≤ iB ≤ iW) ∧ (iR < N))
11              Sinon // LireCouleur(iW) = B
12                  Si (iW ≠ iB) Alors
                    // P5 : P1, LireCouleur(iW) = B, iW ≠ iB
13                      Echanger(iW, iB) ;
14                  FinSi
15                  iB ← iB + 1 ; iW ← iW + 1 ;
                    // P6 : (LireCouleur(iW) = B)_avant → ((1 < iB ≤ iW) ∧ (iR ≤ N))
16              FinSi
                // P7 : (LireCouleur(iW) ≠ W)_avant → ((1 ≤ iB ≤ iW) ∧ (iR ≤ N))
17          FinSi
            // P8 : (1 ≤ iB ≤ iW), (iR ≤ N)
18      FinTantQue
Fin
```

De façon élémentaire, on a :
Résultat 1 : $P_1 \vDash P_2$.
Résultat 2 : $P_1 \vDash P_3$.
Résultat 3 : $P_1 \vDash P_4$.
Résultat 4 : $P_1 \vDash P_5$.
Résultat 5 : $P_1 \vDash P_6$.
Résultat 6 : $\big((1 \leq iB < iW) \wedge (iR \leq N)\big) \vDash \big((1 \leq iB \leq iW) \wedge (iR \leq N)\big)$.
Résultat 7 : $\big((1 \leq iB \leq iW) \wedge (iR < N)\big) \vDash \big((1 \leq iB \leq iW) \wedge (iR \leq N)\big)$.
Résultat 8 : $\big((1 < iB \leq iW) \wedge (iR \leq N)\big) \vDash \big((1 \leq iB \leq iW) \wedge (iR \leq N)\big)$.
Résultat 9 : $P_1 \vDash P_4 \wedge P_6$
D'après les résultats 7, 8 et 9, on déduit le résultat suivant :
Résultat 10 : $P_1 \vDash P_7$.
Résultat 11 : $P_1 \vDash P_2 \wedge P_7$
D'après les résultats 6 et 11, on déduit le résultat suivant :
Résultat 12 : $P_1 \vDash P_8$.

Il faut maintenant montrer que P_1 est un invariant de boucle, auquel cas, tous les P_i pour $i \in \{2, \ldots, 12\}$ le sont aussi d'après les résultats précédents. Le résultat est facile à admettre sans démonstration formelle, cependant on propose ci-dessous une telle démonstration afin de présenter un raisonnement par récurrence assez simple.

On pose alors le prédicat d'invariant suivant :

$$\forall t \in \mathbb{N}^*, Inv(t) : (t \text{ est un numéro de tour de boucle}) \rightarrow P_1$$

Montrons par récurrence que $Inv(t)$ est vérifiée pour tout $t \in \mathbb{N}^*$.
Initialisation :
P_0 est trivialement vérifié d'après les affectations de la ligne 1. D'autre part, si on entre dans la boucle, c'est qu'à la ligne 2, le test $iW \leq iR$ est VRAI et P_1 est vrai quand on exécute la boucle pour la première fois. Donc $Inv(1)$ est vérifié.

Hérédité :
Il faut montrer que $\forall t \in \mathbb{N}^*, Inv(t) \vDash Inv(t+1)$ (hérédité simple).
On précise ce que signifie $Inv(t+1)$:

$$Inv(t+1) : (t+1 \text{ est un numéro de tour de boucle}) \rightarrow P_1$$

Comme $t \geq 1$, on arrive au $(t+1)^{\text{ème}}$ tour en ayant effectué le $t^{ème}$ tour. Au $t^{ème}$ tour, puisque $Inv(t)$ est vérifié par hypothèse d'hérédité, on en déduit que P_1 est vraie, donc, d'après le résultat 12, P_8 est aussi vraie. Lorsque l'on entre dans le $(t+1)^{\text{ème}}$ tour, le test $(iW \leq iR)$ est VRAI. Donc P_1 est vraie quand on entre dans le $(t+1)^{\text{ème}}$ tour et $Inv(t+1)$ est vérifié.

Conclusion :
$Inv(1)$ est vérifié et $\forall t \in \mathbb{N}^*$, la récurrence simple est vérifiée, donc $\forall t \in \mathbb{N}^*$, $Inv(t)$ est vrai.

Donc à chaque fois que l'on utilise LireCouleur( ) ou Echanger( ), P_1 est vraie. Les préconditions sont donc vérifiées.

Maintenant que nous avons montré que toutes les opérations élémentaires et les appels à fonction s'arrêtent (et donc les tests des lignes 3 et 6), il suffit, pour démontrer l'arrêt, de montrer que la boucle tourne un nombre fini de fois.

Nombre fini de tours de boucle.

On pose le prédicat d'arrêt suivant :

$$\forall k \in \mathbb{N}, PA(k) : (k = iR - iW + 1) \rightarrow (\text{le nbre de tours de boucle est fini})$$

Montrons par récurrence que $PA(k)$ est vérifiée pour tout $k \in \mathbb{N}$.
Initialisation :
Si $k = 0$ alors $iW > iR$ et on n'exécute pas la boucle (cf. test ligne 2). Donc $PA(0)$ est vérifié.

Hérédité :
Il faut montrer que $\forall k \in \mathbb{N}, PA(k) \vDash PA(k+1)$ (hérédité simple).
On précise ce que signifie $PA(k+1)$:

$$PA(k+1) : (k+1 = iR - iW + 1) \rightarrow \text{(le nbre de tours de boucle est fini)}$$

Si $k+1 \neq iR - iW + 1$, alors la prémisse de l'implication est fausse, donc l'implication est vraie et $PA(k+1)$ est trivialement vérifiée.
Nous étudions maintenant le cas où $k + 1 = iR - iW + 1$.
Comme $k \geq 0$, on a $iR - iW + 1 \geq 1$ et on entre dans la boucle. On a 3 cas possibles :

1. $\text{LireCouleur}(iW) = W$ et on exécute la ligne 4 et $iW_{après} = iW + 1$ ($iR_{après} = iR$).
2. $\text{LireCouleur}(iW) = R$ et on exécute la ligne 10 et $iR_{après} = iR - 1$ ($iW_{après} = iW$).
3. $\text{LireCouleur}(iW) = B$ et on exécute la ligne 15 et $iW_{après} = iW + 1$ ($iR_{après} = iR$).

Dans tous les cas on a $iR_{après} - iW_{après} + 1 = iR - iW = k$.
On peut appliquer l'hypothèse d'hérédité $PA(k)$ et donc le nombre de tours pour $k+1$ est fini puisqu'après 1 tour, le nombre de tours pour k est fini. $PA(k+1)$ est donc vrai.

Conclusion :
$PA(0)$ est vérifié et $\forall k \in \mathbb{N}$, l'hérédité simple est vérifiée, donc $\forall k \in \mathbb{N}, PA(k)$ est vrai.

Le nombre de tours de boucle est donc fini et, puisque chaque instruction élémentaire et chaque appel de fonction s'arrête, la fonction s'arrête.

2. Démonstration du bon résultat

On rappelle les trois prédicats portant sur les couleurs :

- $P_B : \forall i, (\, i \in \mathbb{N} \cap [1, iB[) \rightarrow (\text{LireCouleur}(i) = B)$
- $P_W : \forall i, (\, i \in \mathbb{N} \cap [iB, iW[) \rightarrow (\text{LireCouleur}(i) = W)$
- $P_R : \forall i, (\, i \in \mathbb{N} \cap]iR, N]) \rightarrow (\text{LireCouleur}(i) = R)$

Et on note $P_C : P_B \wedge P_W \wedge P_R$.
On note $P'_C : P_B \wedge P'_W \wedge P_R$, où P'_W est $\forall i, (\, i \in \mathbb{N} \cap [iB + 1, iW[) \rightarrow (\text{LireCouleur}(i) = W)$.

D'autre part, on note P_N : $(|T|_B = \alpha) \wedge (|T|_W = \beta) \wedge (|T|_R = \gamma)$ où α, β et γ désignent respectivement le nombre de pierres de couleur bleue, blanche et rouge initialement sur la table.
Comme aucune action ne modifie la couleur d'une pierre et n'enlève ni n'ajoute de pierre, on a le résultat suivant :
Résultat 13 : P_N est invariant tout au long de l'algorithme.

Afin de ne pas surcharger la notation, P_N ne figurera de façon explicite que dans les prédicats $P_{initial}$ et P_{final}. De même, on fera figurer les inégalités entre indices établies lors de la démonstration de l'arrêt uniquement dans P_{final}.

Variables
  entier iB, iW, iR ;
Début
1 iB ← 1 ; iW ← 1 ; iR ← N ;
   // $P_{initial} : P_C, P_N$
2 Tant Que (iW ≤ iR) Faire
   // $PDB : P_C$
3   Si (LireCouleur(iW) = W) Alors
     // $PIB_0 : P_C, (\text{LireCouleur}(iW) = W)$

4 iW ← iW + 1 ;
// PIB_1 : $(\text{LireCouleur}(iW) = W)_{avant} \rightarrow P_C$
5 Sinon
// PIB_2 : $P_C, (\text{LireCouleur}(iW) \neq W)$
6 Si (LireCouleur(iW) = R) Alors
// PIB_3 : $P_C, (\text{LireCouleur}(iW) = R)$
7 SI (iW ≠ iR) Alors
// PIB_4 : $P_C, \text{LireCouleur}(iW) = R, iW \neq iR$
8 Echanger(iW, iR) ;
// PIB_5 : $P_C, \text{LireCouleur}(iR) = R, iW \neq iR$
9 FinSi
// PIB_6 : $P_C, \text{LireCouleur}(iR) = R$
10 iR ← iR – 1 ;
// PIB_7 : $(\text{LireCouleur}(iW) = R)_{avant} \rightarrow P_C$
11 Sinon // LireCouleur(iW) = B
// PIB_8 : $P_C, (\text{LireCouleur}(iW) = B)$
12 Si (iW ≠ iB) Alors
// PIB_9 : $P_C, \text{LireCouleur}(iW) = B, iW \neq iB$
13 Echanger(iW, iB) ;
/* PIB_{10} : $P'_C, \text{LireCouleur}(iB) = B, \text{LireCouleur}(iW) = W,$
$iW \neq iB$ */
14 FinSi
/* PIB_{11} : $P'_C, \text{LireCouleur}(iB) = B,$
$(iW \neq iB) \rightarrow (\text{LireCouleur}(iW) = W)$ */
15 iB ← iB + 1 ; iW ← iW + 1 ;
// PIB_{12} : $(\text{LireCouleur}(iW) = B)_{avant} \rightarrow P_C$
16 FinSi
// PIB_{13} : $(\text{LireCouleur}(iW) \neq W)_{avant} \rightarrow P_C$
17 FinSi
// PFB : P_C
18 FinTantQue
// P_{final} : $P_C, P_N, 1 \leq iB \leq iW, iW = iR + 1, iR \leq N$
Fin

Résultat 14 : $P_{initial}$ est vrai.
On a vu que P_N est vrai, de plus, P_C est vérifié par défaut car les trois intervalles $[1, iB[$, $[iB, iW[$ et $]iR, N]$ sont vides, donc les prédicats P_B, P_W et P_R sont vrais puisque leur prémisse est fausse. Donc $P_{initial}$ est vrai.

Résultat 15 : Au premier tour de boucle, PDB est vrai.
Ce résultat est trivial : PDB et $P_{initial}$ sont les mêmes prédicats (on omet P_N) et puisqu'aucune variable n'a changé de valeur au premier tour, PDB est vrai d'après le résultat 14.

Le résultat suivant est immédiat :
Résultat 16 : PIB_0 se déduit de PDB et du test de la ligne 3.

À la ligne 4, seule iW change de valeur, il suffit alors de vérifier que P_W est encore vrai après la ligne 4 pour démontrer que PIB_1 se déduit bien de PIB_0. Soit $iW_{après}$ la valeur de iW après la ligne 4 : $iW_{après} = iW + 1$. $(\text{LireCouleur}(iW) = W)_{avant}$ signifie que la pierre en iW avant modification de iW est blanche. Donc, avant modification, on a :
$\forall i, (\, i \in \mathbb{N} \cap [iB, iW[) \rightarrow (\text{LireCouleur}(i) = W)$ et $\text{LireCouleur}(iW) = W$, on peut donc en déduire (en intégrant la connaissance de la couleur en iW dans P_W) :

$$\forall i, (\, i \in \mathbb{N} \cap [iB, iW + 1[) \rightarrow (\text{LireCouleur}(i) = W)$$

ou encore :

$$\forall i, \left(i \in \mathbb{N} \cap \left[iB, iW_{après}\right[\right) \rightarrow (\text{LireCouleur}(i) = W)$$

donc P_W est vrai après l'exécution de la ligne 4. On a donc le résultat :
Résultat 17 : $\left(PIB_0 \wedge (\text{LireCouleur}(iW) = W)\right) \vDash PIB_1$.

Les résultats suivants sont immédiats :
Résultat 18 : $\left(PIB_0 \wedge (\text{LireCouleur}(iW) \neq W)\right) \vDash PIB_2$.
Résultat 19 : $\left(PIB_2 \wedge (\text{LireCouleur}(iW) = R)\right) \vDash PIB_3$.
Résultat 20 : $\left(PIB_3 \wedge (iW \neq iR)\right) \vDash PIB_4$.

Puisque les pierres en iW et en iR ont été échangées à la ligne 8, on a :
Résultat 21 : $PIB_4 \vDash PIB_5$.

Après la ligne 9, on a : $\left(PIB_3 \wedge (iW = iR)\right) \rightarrow PIB_6$ et $\left(PIB_3 \wedge (iW \neq iR)\right) \rightarrow PIB_6$ donc :
Résultat 22 : $PIB_3 \vDash PIB_6$.

Par un raisonnement similaire à celui utilisé pour le résultat 17, on a :
Résultat 23 : $\left(PIB_2 \wedge (\text{LireCouleur}(iW) = R)\right) \vDash PIB_7$.

Le « Sinon » de la ligne 11 correspond à $(\text{LireCouleur}(iW) \neq W) \wedge (\text{LireCouleur}(iW) \neq R)$, la seule couleur restant étant B, on a bien :
Résultat 24 : $\left(PIB_2 \wedge (\text{LireCouleur}(iW) \neq R)\right) \vDash PIB_8$.

Le résultat suivant est immédiat :
Résultat 25 : $\left(PIB_8 \wedge (iW \neq iB)\right) \vDash PIB_9$.

Comme $iW \neq iB$, on a $(\text{LireCouleur}(iB) = W)$ (voir P_W) et puisque les pierres en iW et en iB ont été échangées à la ligne 13, on a :
Résultat 26 : $PIB_9 \vDash PIB_{10}$.

Après la ligne 14, on a : $\left(PIB_8 \wedge (iW = iB)\right) \rightarrow \left(P'_C \wedge (\text{LireCouleur}(iB) = B)\right)$ et $\left(PIB_8 \wedge (iW = iB)\right) \rightarrow \left(P'_C \wedge (\text{LireCouleur}(iB) = B) \wedge (\text{LireCouleur}(iW) = W)\right)$ donc :
Résultat 27 : $PIB_8 \vDash PIB_{11}$.

Pour démontrer que l'on peut déduire PIB_{12} de PIB_{11}, on applique le même raisonnement que pour le résultat 17 mais cette fois-ci sur deux prédicats : P_B et P_W ou plus exactement sur P_B et P'_W. À la ligne 15, ce sont iB et iW qui changent de valeur. On note $iB_{après}$ et $iW_{après}$ les valeurs de iB et iW après la ligne 15 : $iB_{après} = iB + 1$ et $iW_{après} = iW + 1$. $(\text{LireCouleur}(iW) = B)_{avant}$ signifie que la pierre en iW avant modification de iW est bleue. Donc, avant modification, on a : $\forall i, (i \in \mathbb{N} \cap [1, iB[) \rightarrow (\text{LireCouleur}(i) = B)$, $\forall i, (i \in \mathbb{N} \cap [iB, iW[) \rightarrow (\text{LireCouleur}(i) = W)$ et $\text{LireCouleur}(iW) = B$, on peut donc en déduire (en intégrant la connaissance de la couleur en iB dans P_B) :

$$\forall i, (i \in \mathbb{N} \cap [1, iB + 1[) \rightarrow (\text{LireCouleur}(i) = B)$$

ou encore :

$$\forall i, \left(i \in \mathbb{N} \cap \left[1, iB_{après}\right[\right) \rightarrow (\text{LireCouleur}(i) = B)$$

donc P_B est vrai après l'exécution de la ligne 15.
On a deux cas à discuter pour établir P_W :

- $iW = iB$:
  dans ce cas $iB_{après} = iW_{après}$ et $\left[iB_{après}, iW_{après}\right[= [iB, iW[= \emptyset$ et P_W est vrai.
- $iW \neq iB$

dans ce cas il y a eu échange et LireCouleur$(iW) = W$ et en intégrant ce prédicat à P'_W, on obtient $\forall i, (\, i \in \mathbb{N} \cap [iB + 1, iW + 1[) \rightarrow$ (LireCouleur$(i) = W$) qui peut s'écrire aussi $\forall i, \left(\, i \in \mathbb{N} \cap \left[iB_{après}, iW_{après}\right[\right) \rightarrow$ (LireCouleur$(i) = W$) qui est bien P_W.

Dans les deux cas, P_W est vrai. On peut donc en déduire :
Résultat 28 : $PIB_{11} \vDash PIB_{12}$.

On a donc le résultat :
Résultat 29 : $(PIB_7 \wedge PIB_{12}) \vDash PIB_{13}$.

De même, on a :
Résultat 30 : $(PIB_1 \wedge PIB_{13}) \vDash PFB$.

Et comme PIB_1 et PIB_{13} se déduisent de PDB (par transitivité de $\vDash$) on a :
Résultat 31 : Dans un tour de boucle $PDB \vDash PFB$.

On pose alors le prédicat suivant :

$$\forall t \in \mathbb{N}^*, P(t) : (t \text{ est un numéro de tour de boucle}) \rightarrow PFB(t)$$

où $PFB(t)$ signifie que PFB est vérifié dans le $t^{ième}$ tour de boucle.

Montrons par récurrence que $P(t)$ est vérifié pour tout $t \in \mathbb{N}^*$.
Initialisation :
D'après le résultat 15, $PDB(1)$ est vérifié au premier tour de boucle. D'après le résultat 31, $PFB(1)$ est vérifié dans ce même tour de boucle. Donc $P(1)$ est vrai.

Hérédité :
Il faut montrer : $\forall t \in \mathbb{N}^*, P(t) \vDash P(t+1)$ (hérédité simple).
On précise ce que signifie $P(t+1)$:

$$P(t+1) : (t+1 \text{ est un numéro de tour de boucle}) \rightarrow PFB(t+1)$$

Comme $t \geq 1$ on arrive au $(t+1)^{ème}$ tour en ayant effectué le $t^{ème}$ tour. Au $t^{ème}$ tour, puisque $P(t)$ est vraie par hypothèse d'hérédité, on en déduit que $PFB(t)$ est vérifié à la fin de ce tour.
Au $(t+1)^{ème}$ tour, le test du « TantQue » est VRAI, donc $PDB(t+1)$ est vrai. D'après le résultat 31, $PFB(t+1)$ est donc vrai et $P(t+1)$ est vrai.

Conclusion :
$P(1)$ est vrai et $\forall t \in \mathbb{N}^*$, l'hérédité simple est vérifiée, donc $\forall t \in \mathbb{N}^*$, $P(t)$ est vrai.

Donc PFB est un invariant de boucle.

Il faut maintenant montrer que P_{final} est vrai. On rappelle que P_{final} est le prédicat suivant :

$$P_C \wedge P_N \wedge (1 \leq iB \leq iW) \wedge (iW = iR + 1) \wedge (iR \leq N)$$

1. On suppose tout d'abord que l'on a effectué au moins un tour de boucle. Lorsque la boucle s'arrête, le test de la ligne 2 est faux et on a comme prédicat en sortie de boucle : $P_C \wedge (iW = iR + 1)$ auquel on peut adjoindre P_8 (cf. démonstration de l'arrêt) et P_N (cf. résultat 13). On a bien P_{final}.
2. On suppose maintenant que la boucle n'est pas effectuée, cela signifie que l'on a $P_{initial} \wedge (iW = iR + 1)$ on a donc $P_C \wedge P_N \wedge (iW = iR + 1)$ auquel on peut adjoindre P_8 comme dans le cas 1. On a bien P_{final}.

Dans les deux cas P_{final} est vrai.

De P_B, on peut déduire que $T' \in B^*E$ où $E \subseteq C^*$, de P_W, on peut déduire que $T' \in B^*W^*E$ et de P_R, on peut déduire que $T' \in B^*W^*ER^*$. De $iW = iR + 1$, on peut déduire que $E = \emptyset$ et que $T' \in B^*W^*R^*$. De P_N, on peut déduire que $(|T'|_B = |T|_B) \wedge (|T'|_W = |T|_W) \wedge (|T'|_R = |T|_R)$ et que finalement $T' \in C^N \cap B^*W^*R^*$. Donc on a bien la postcondition. La fonction renvoie bien le résultat attendu.

COMPLEXITÉ EN TEMPS

Toutes les opérations élémentaires et les appels à fonction sont en temps constant.

On pose le prédicat sur le nombre de tours de boucles suivant :

$$\forall k \in \mathbb{N}, PNT(k) : (k = iR - iW + 1) \rightarrow (\#\text{tours} = k)$$

où #tours désigne le nombre de tours de boucles.

Montrons par récurrence que $PNT(k)$ est vérifiée pour tout $k \in \mathbb{N}$.

Initialisation :

Si $k = 0$ alors $iW > iR$ et on n'exécute pas la boucle (cf. test ligne 2). Donc $PNT(0)$ est vrai.

Hérédité :

Il faut montrer que $\forall k \in \mathbb{N}$, $PNT(k) \vDash PNT(k+1)$ (hérédité simple).

On précise ce que signifie $PNT(k+1)$:

$$PNT(k+1) : (k + 1 = iR - iW + 1) \rightarrow (\#\text{tours} = k + 1)$$

Si $k + 1 \neq iR - iW + 1$, alors la prémisse de l'implication est fausse, donc l'implication est vraie et $PNT(k+1)$ est trivialement vérifiée.

Nous étudions maintenant le cas où $k + 1 = iR - iW + 1$.

Comme $k \geq 0$, on a $iR - iW + 1 \geq 1$ et on entre dans la boucle. On a 3 cas possibles :

1. LireCouleur$(iW) = W$ et on exécute la ligne 4 et $iW_{après} = iW + 1$ ($iR_{après} = iR$).
2. LireCouleur$(iW) = R$ et on exécute la ligne 10 et $iR_{après} = iR - 1$ ($iW_{après} = iW$).
3. LireCouleur$(iW) = B$ et on exécute la ligne 15 et $iW_{après} = iW + 1$ ($iR_{après} = iR$).

Dans tous les cas on a $iR_{après} - iW_{après} + 1 = iR - iW = k$.

Donc le nombre de tours pour $k + 1$ est égal à 1, pour passer de $k + 1$ à k et à partir de k on peut appliquer l'hypothèse d'hérédité $PNT(k)$ ce qui donne k tours. On a donc $\#\text{tours} = k + 1$ et $PNT(k+1)$ est vrai.

Conclusion :

$PNT(0)$ est vérifié et $\forall k \in \mathbb{N}$, l'hérédité simple est vérifiée, donc $\forall k \in \mathbb{N}$, $PNT(k)$ est vrai.

Puisque toutes les opérations sont en temps constant, la complexité en temps pour une table à N pierres est donc en $\theta(N)$.