

Compte rendu

Projet Hanoï

GRAVE Valentin

①

Sur une tour, pour pouvoir empiler un disque sur un autre, il est nécessaire de satisfaire les critères suivants :

→ Le diamètre du disque (typé "Int") à empiler est une valeur entière comprise entre 1 et 2 147 483 647 (car le type "Int" permet cette valeur positive au maximum)

(doit être validé, quelle que soit la situation qui suit)

- Si la tour est vide :

→ N'importe quel disque peut être ajouté

- Si la tour n'est pas vide :

→ Le disque à empiler a un diamètre inférieur au diamètre du disque du sommet de la tour

→ La capacité (en disques) de la tour n'est pas atteinte

→ Un disque ayant le diamètre d'un disque déjà présent dans le système ne peut pas être ajouté

Ce qui peut se traduire par le tableau suivant : ("diam()") représente le diamètre du cercle considéré)

| Paramètre d'entrée | Classe valide | Classe invalide |
|---|---|--|
| Empiler un disque D (tour T vide) | ($0 < \text{D.diam}() < 2\,147\,483\,648$) | ($\text{D.diam}() \leq 0$) OU ($\text{D.diam}() \geq 2\,147\,483\,648$) |
| Empiler un disque D (tour T non vide, avec au sommet un disque S) | ($0 < \text{D.diam}() < 2\,147\,483\,648$) ET ($\text{D.diam}() < \text{S.diam}()$) ET ($\text{T.taille}() < \text{T.tailleMax}()$) ET ($\text{D.diam}()$ non présent parmi les diam() des disques du système) | ($\text{D.diam}() \leq 0$) OU ($\text{D.diam}() \geq 2\,147\,483\,648$) OU ($\text{D.diam}() \geq \text{S.diam}()$) OU ($\text{T.taille}() \geq \text{T.tailleMax}()$) OU ($\text{D.diam}()$ présent parmi les diam() des disques du système) |

②

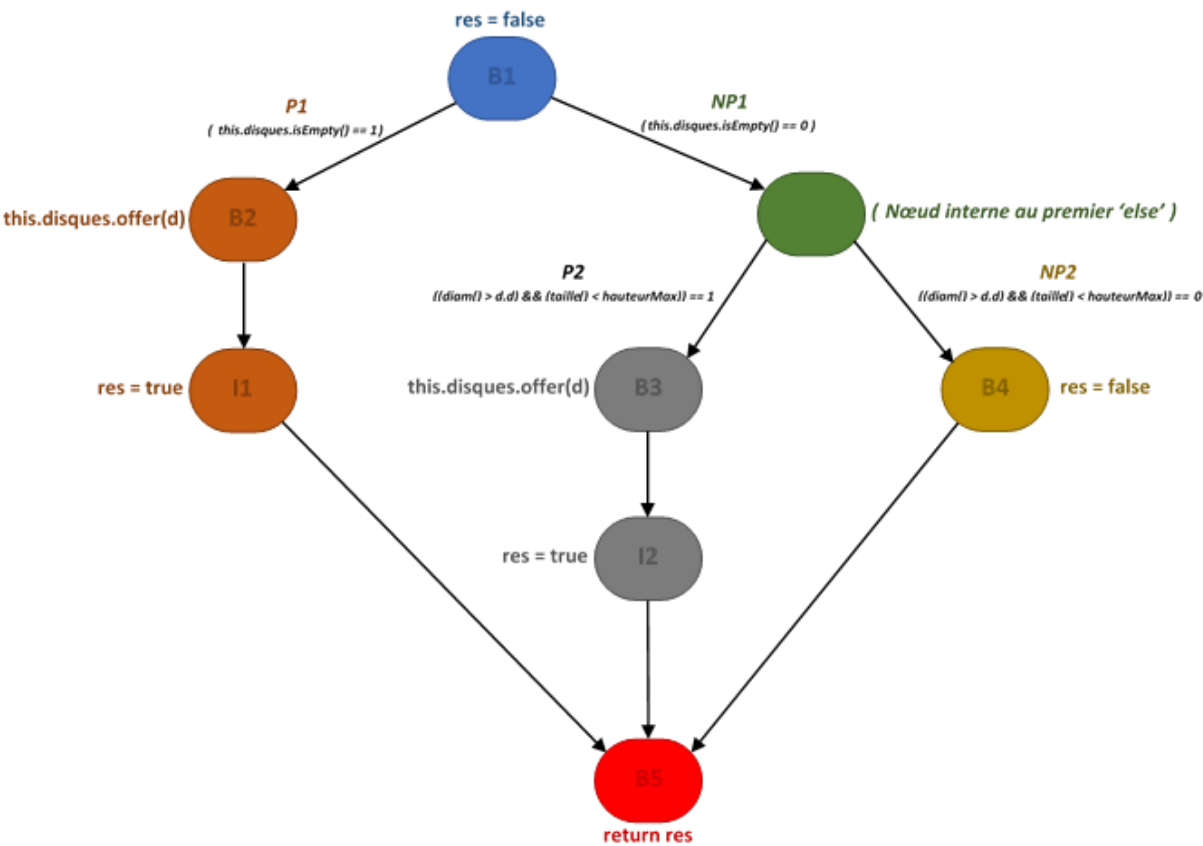
Voici les différents tests aux limites qu'il faudrait effectuer, en se référant au tableau de la

question ① : (en sachant que les diamètres des disques sont forcément des entiers, selon le constructeur de la classe Disque)

| Paramètre d'entrée | Classe valide | Classe invalide |
|---|--|---|
| Empiler un disque D (tour T vide) | ($\text{D.diam}() = 1$) OU ($\text{D.diam}() = 2\,147\,483\,647$) | ($\text{D.diam}() = -1$) OU ($\text{D.diam}() = 0$) OU ($\text{D.diam}() = 2\,147\,483\,648$) |
| Empiler un disque D (tour T non vide, avec au sommet un disque S) | (($\text{D.diam}() = 1$) OU ($\text{D.diam}() = 2\,147\,483\,647$)) ET ($\text{D.diam}() = \text{S.diam}() - 1$) ET ($\text{T.taille}() = \text{T.tailleMax}() - 1$) ET ($\text{D.diam}()$ non présent parmi les diam() des disques du système) | (($\text{D.diam}() = -1$) OU ($\text{D.diam}() = 0$) OU ($\text{D.diam}() = 2\,147\,483\,648$)) OU (($\text{D.diam}() = \text{S.diam}()$) OU ($\text{D.diam}() = \text{S.diam}() + 1$)) OU (($\text{T.taille}() = \text{T.tailleMax}()$) OU ($\text{T.taille}() = \text{T.tailleMax}() + 1$)) OU ($\text{D.diam}()$ présent parmi les diam() des disques du système) |

③

Ci-dessous se trouve le graphe de flot de contrôle correspondant à la fonction "empiler()".
 Veuillez noter que je me suis permis d'ajouter un point //NP1 à la suite du premier "else", en effet, il me semblait pertinent d'ajouter ce point qui est l'alternative à //P1 si le premier test "if" n'est pas validé.
 Similairement, j'ai rajouté //NP2 en guise d'alternative à //P2 si le second test "if" n'est pas satisfait.



Voici la forme algébrique de ce graphe de flot de contrôle :

ensemble_des_chemins_possibles = B1.B2.I1.B5 + B1.B3.I2.B5 + B1.B4.B5

- chemins_menant_à_res_true = B1.B2.I1.B5 + B1.B3.I2.B5
- chemins_menant_à_res_false = B1.B4.B5

④

D'après le graphe de flot de contrôle réalisé ci-dessus, il serait nécessaire de tester les arcs avec les données de test suivantes :

ATTENTION :
 Pour juste parcourir tous les instructions/arcs du graphe (sans en explorer toutes les possibilités) les 3 premiers points suivants suffisent.
 Les 2 autres points ont été ajoutés car je n'étais pas sûr qu'il était suffisant de seulement parcourir le graphe, j'ai préféré détailler tous les scénarios possibles, bien que je crois que ceux-ci soient plutôt les réponses à la question ⑥.

- **Transition P1 :**
 une tour vide et un disque
 → Devrait retourner : res == true
- **Transitions NP1 et P2 :**
 une tour non vide (mais avec au moins une place libre) et un disque, de diamètre inférieur au disque du sommet de la tour
 → Devrait retourner : res == true
- **Transitions NP1 et NP2 :**
 une tour non vide (mais avec au moins une place libre) et un disque, de diamètre supérieur (ou égal) au disque du sommet de la tour
 → Devrait retourner : res == false
- **Transitions NP1 et NP2 :**
 une tour non vide (mais sans place libre) et un disque, de diamètre inférieur au disque du sommet de la tour
 → Devrait retourner : res == false
- **Transitions NP1 et NP2 :**
 une tour non vide (mais sans place libre) et un disque, de diamètre supérieur (ou égal) au disque du sommet de la tour
 → Devrait retourner : res == false

⑤

Toutes les arcs ont été parcourus dans la question ④, mais des tests essentiels au bon fonctionnement du programme ne sont pas effectués par le graphe de flot de contrôle de la fonction "empiler()" :

- Ajouter, préférablement au constructeur de la classe **Disque**, les vérifications concernant le diamètre d'un disque, afin que celui-ci soit un entier compris **STRICTEMENT** entre 0 et 2 147 483 648.

En effet, il n'y a pas de disque si son diamètre est de 0, un "Int" ne va pas au-delà de 2 147 483 647 et un diamètre négatif résulterait en un opposé de disque ?

Il faut donc veiller à faire ce test préliminaire (afin de ne pas perdre du temps à tenter d'ajouter un disque invalide à la tour).

- Ajouter un test à la fonction "empiler()" regroupant les diamètres des disques déjà présents au sein du système, afin qu'un disque ayant un diamètre déjà représenté dans la liste ne puisse pas être ajouté.

⑥

Comme dit précédemment, à la question ④, voici l'ensemble des tests complet à effectuer pour la condition composée **((diam() > d.d) && (taille() < hauteurMax))** :

- Transitions NP1 et P2 :
une tour non vide (mais avec au moins une place libre) et un disque, de diamètre inférieur au disque du sommet de la tour
→ Devrait retourner : res == true
- Transitions NP1 et NP2 :
une tour non vide (mais avec au moins une place libre) et un disque, de diamètre supérieur (ou égal) au disque du sommet de la tour
→ Devrait retourner : res == false
- Transitions NP1 et NP2 :
une tour non vide (mais sans place libre) et un disque, de diamètre inférieur au disque du sommet de la tour
→ Devrait retourner : res == false
- Transitions NP1 et NP2 :
une tour non vide (mais sans place libre) et un disque, de diamètre supérieur (ou égal) au disque du sommet de la tour
→ Devrait retourner : res == false

⑦

(Voir code TourTest.java)

⑧

(Voir code DisqueTest.java)

⑨

(Voir code Hanoi.java)