IUT Sénart-Fontainebleau, UPEC
DEV 2.3
IUT Info 1A

Florent Madelaine
Luc Dartois
Luc Hernandez
2023-24

Quels tests ?

Notions importantes

- tests unitaires
- couverture (méthode, chemin etc)

Exercices

Exercice 1. On considère l'algorithme suivant. Le membre de votre équipe de développement vous demande de le tester pour lui et vous indique que cette méthode permet de savoir si un entier est premier.

```
public boolean isPrime(int number) {
   if (number == 1)
      return false;
   if (number == 2)
      return true;
   if (number % 2 == 0)
      return false;
   for (int d = 3; d <= (int) Math.sqrt(number); d++)=
      if (number % d == 0)
         return false;
   return true;
}
```

1. Quelles sont les préconditions et postconditions de cette méthode ?
2. Proposez un test exhaustif (NB. 2147483647 est le plus grand int en java).
3. Donnez un algorigramme / diagramme d'activité pour

   ```
   if(test)
      then
   suite
   ```

4. Même question pour

   ```
   for(init;test;incr)
      truc
   suite
   ```

5. Construisez un algorigramme / diagramme d'activité pour cette méthode `isPrime`.
6. Proposez un premier jeu de tests unitaires permettant de couvrir toutes les branches de cette méthode.
7. Proposez des tests unitaires aléatoires permettant de vérifier de nombreux cas où la réponse devrait être False.

8. La méthode de Miller Rabin est une méthode aléatoire qui se trompe parfois mais permet de vérifier que des nombres sont probablement premiers. En supposant qu'un autre membre de l'équipe nous donne son code pour sa méthode `IsPrimeMillerRabin(int number)`, quels tests unitaires proposez vous ?

Exercice 2. On souhaite faire des tests en boîte noire de certains algorithmes. Proposez pour chaque algorithme un banc de test adapté (*benchmarks*).

1. **Connexité**. L'algorithme prend un graphe en entrée et demande si ce graphe est connexe. Le graphe doit être simple et sans boucle. Le graphe est donné par une matrice d'adjacence.
2. **Degré**. L'algorithme prend en entrée un graphe simple et sans boucle donné par une matrice d'adjacence et un entier n. Il doit répondre vrai si le graphe contient un sommet avec n voisins.
3. **Racine entière**. L'algorithme prend en entrée un entier n et retourne $\lfloor\sqrt{n}\rfloor$ la partie entière inférieure de la racine carré de n.

Exercice 3. On suppose qu'on dispose de pièces et billets dans notre caisse. Le client paye trop et on doit lui rendre sa monnaie.

1. Décrivez la méthode gloutonne qu'on appellera « méthode de la boulangère » sur des exemples.
2. Donnez le pseudo-code de cette méthode
3. Dessinez un diagramme d'activité.

Pour aller plus loin

Exercice 4. On se donne une suite de $r \geq 1$ entiers strictement positifs p_1,p_2,...,p_r. On se donne un booléen pour chaque entier de la suite, soit une suite $0 \leq b_1,b_2,...,b_r \leq 1$. Pour un tel choix on calcule la somme : $S = b_1 \times p_1 + b_2 \times p_2 ... b_r \times p_r$

Intuition. p_i est le poids d'un objet i. S est le poids du sac à dos contenant l'objet i exactement lorsque b_i vaut 1.

On joue au jeu suivant. Le premier joueur cache des objets dans le sac à dos. Le second joueur connaît le poids de chaque objet et celui du sac à dos. Il doit deviner sans regarder dans le sac, les objets que ce dernier contient.

Cette suite est hyper-croissante ssi pour tout $1 < j \leq r$, la somme $p_1 + ... + p_{j-1} \leq p_j$.

1. Vérifier que 1,2,4,8,16,32 est une suite hyper croissante.
2. Vérifier que 1,5,10,50,100,200,500,1000 est une suite hyper croissante.
3. Montrez que si la suite des poids est hypercroissante alors on peut utiliser un *algorithme glouton* pour retrouver les poids à partir de S.