Sections 5.3 et 5.4

1. Calculer les premières valeurs de chacune des fonctions suivantes.

   (a)
   $$f(n) = \begin{cases} 1 & \text{si } n = 1, \\ f(n-1) & \text{sinon.} \end{cases}$$

   (b)
   $$f(n) = \begin{cases} 20 & \text{si } n = 1, \\ \frac{1}{2} f(n-1) & \text{si } f(n-1) \text{ est pair,} \\ 3f(n-1) + 1 & \text{si } f(n-1) \text{ est impair.} \end{cases}$$

   (c)
   $$f(n) = \begin{cases} 1 & \text{si } n = 1, \\ n \cdot f(n-1) & \text{sinon.} \end{cases}$$

   (d)
   $$f(n) = \begin{cases} 1 & \text{si } n = 1, \\ 11 \cdot f(n-1) & \text{sinon.} \end{cases}$$

   (e)
   $$f(n) = \begin{cases} 1 & \text{si } n = 1, \\ 2 & \text{si } n = 2, \\ \frac{f(n-2)}{f(n-1)} & \text{sinon.} \end{cases}$$

2. Trouver une formule pour chacune des fonctions de la question précédente et démontrer que votre réponse est correcte.

3. Définir récursivement chacun des ensembles suivants. Démontrer que votre réponse est correcte.
   (a) L'ensemble des entiers positifs pair.
   (b) L'ensemble des multiples de 5 positifs.
   (c) L'ensemble des puissances de 3.
   (d) L'ensemble des entiers qui sont des carrés parfaits.

4. Quel est l'ensemble défini récursivement par
   — $S \subseteq \mathbb{Z}^2$
   — $(0, 0) \in S$
   — Si $(a, b) \in S$, alors $(a, b + 1) \in S$.
   — Si $(a, b) \in S$, alors $(a + 1, b + 1) \in S$.

Démontrer que votre réponse est correcte.

5. Quel est l'ensemble défini récursivement par
   — $S \subseteq \mathbb{Z}^+ \times \mathbb{Z}^+$
   — $(0,1) \in S$
   — $(1,0) \in S$
   — Si $(a,b) \in S$, alors $(a+2,b) \in S$.
   — Si $(a,b) \in S$, alors $(a,b+2) \in S$.

   Démontrer que votre réponse est correcte.
6. Quel est l'ensemble défini récursivement par
   — $8 \in S$
   — $6 \in S$
   — Si $a,b \in S$, alors $a+b \in S$
   — Si $a,b \in S$, alors $a-b \in S$

   Démontrer que votre réponse est correcte.
7. Écrire le pseudocode pour la recherche dichotomique (binary search), version récursive.
   — Identifier clairement le cas de base.
   — Indiquer en combien de sous-problèmes on divise l'input.
   — Indiquer combien d'appels récursifs on fait avec ces sous-problèmes.
   — Quel est le merge step ?
8. Écrire le pseudocode pour la recherche séquentielle, version récursive.
   — Identifier clairement le cas de base.
   — Indiquer en combien de sous-problèmes on divise l'input.
   — Indiquer combien d'appels récursifs on fait avec ces sous-problèmes.
   — Quel est le merge step ?
9. Concevoir un algorithme récursif qui calcule le nombre de « 1 » dans une chaîne de caractères binaires.
   — Identifier clairement le cas de base.
   — Indiquer en combien de sous-problèmes on divise l'input.
   — Indiquer combien d'appels récursifs on fait avec ces sous-problèmes.
   — Quel est le merge step ?
10. Concevoir un algorithme récursif qui calcule la longueur de la plus longue suite de « 1 » consécutifs dans une chaîne de caractères binaires.
    — Identifier clairement le cas de base.
    — Indiquer en combien de sous-problèmes on divise l'input.

- Indiquer combien d'appels récursifs on fait avec ces sous-problèmes.
- Quel est le merge step ?

11. Concevoir un algorithme récursif qui calcule la longueur de la plus longue suite de la forme ≪ ..., 0, 1, 0, 1, 0, 1, 0, 1, 0, ... ≫ dans une chaîne de caractères binaires.
    - Identifier clairement le cas de base.
    - Indiquer en combien de sous-problèmes on divise l'input.
    - Indiquer combien d'appels récursifs on fait avec ces sous-problèmes.
    - Quel est le merge step ?
12. Concevoir un algorithme récursif qui calcule le plus petit élément d'un tableau de nombres.
    - Identifier clairement le cas de base.
    - Indiquer en combien de sous-problèmes on divise l'input.
    - Indiquer combien d'appels récursifs on fait avec ces sous-problèmes.
    - Quel est le merge step ?
13. Concevoir un algorithme récursif qui calcule le deuxième plus petit élément d'un tableau de nombres.
    - Identifier clairement le cas de base.
    - Indiquer en combien de sous-problèmes on divise l'input.
    - Indiquer combien d'appels récursifs on fait avec ces sous-problèmes.
    - Quel est le merge step ?
14. Concevoir un algorithme récursif qui calcule le plus grand commun diviseur de deux nombres.
    - Identifier clairement le cas de base.
    - Indiquer en combien de sous-problèmes on divise l'input.
    - Indiquer combien d'appels récursifs on fait avec ces sous-problèmes.
    - Quel est le merge step ?
15. Concevoir un algorithme récursif qui calcule le plus petit commun multiple de deux nombres.
    - Identifier clairement le cas de base.
    - Indiquer en combien de sous-problèmes on divise l'input.
    - Indiquer combien d'appels récursifs on fait avec ces sous-problèmes.
    - Quel est le merge step ?
16. Concevoir un algorithme récursif qui calcule le $n^{\text{ième}}$ nombre de Fibonacci.
    - Identifier clairement le cas de base.
    - Indiquer en combien de sous-problèmes on divise l'input.

- Indiquer combien d'appels récursifs on fait avec ces sous-problèmes.
- Quel est le merge step ?

17. Concevoir un algorithme récursif qui calcule la somme de tous les éléments d'un tableau de nombres.
    - Identifier clairement le cas de base.
    - Indiquer en combien de sous-problèmes on divise l'input.
    - Indiquer combien d'appels récursifs on fait avec ces sous-problèmes.
    - Quel est le merge step ?
18. Concevoir un algorithme récursif qui calcule la moyenne de tous les éléments d'un tableau de nombres.
    - Identifier clairement le cas de base.
    - Indiquer en combien de sous-problèmes on divise l'input.
    - Indiquer combien d'appels récursifs on fait avec ces sous-problèmes.
    - Quel est le merge step ?