Chapitre n$^{\text{o}}$ 11 : exercices

1

Il y a 5 passages dans la boucle extérieure, i prenant les valeurs 0, 1, 2, 3 et 4.

Lorsque i vaut 0, la boucle intérieure a 5 passages, j prenant les valeurs 0, 1, 2, 3 et 4.
Il y a donc 5 appels de la fonction *print*.

Lorsque i vaut 1, la boucle intérieure a 4 passages, j prenant les valeurs 1, 2, 3 et 4.
Il y a donc 4 appels de la fonction *print*.

Lorsque i vaut 2, la boucle intérieure a 3 passages, j prenant les valeurs 2, 3 et 4.
Il y a donc 3 appels de la fonction *print*.

Lorsque i vaut 3, la boucle intérieure a 2 passages, j prenant les valeurs 3 et 4.
Il y a donc 2 appels de la fonction *print*.

Lorsque i vaut 4, la boucle intérieure a 1 passage, j prenant la valeurs 4.
Il y a donc 1 appel de la fonction *print*.

$5+4+3+2+1=15$, donc **il y a au total 15 appels à la fonction *print***.

2

Il y a 5 passages dans la boucle extérieure, i prenant les valeurs 0, 1, 2, 3 et 4.

Lorsque i vaut 0, la boucle intérieure a 4 passages, j prenant les valeurs 1, 2, 3 et 4.
Il y a donc 4 appels de la fonction *print*.

Lorsque i vaut 1, la boucle intérieure a 3 passages, j prenant les valeurs 2, 3 et 4.
Il y a donc 3 appels de la fonction *print*.

Lorsque i vaut 2, la boucle intérieure a 2 passages, j prenant les valeurs 3 et 4.
Il y a donc 2 appels de la fonction *print*.

Lorsque i vaut 3, la boucle intérieure a 1 passage, j prenant la valeurs 4.
Il y a donc 1 appel de la fonction *print*.

Lorsque i vaut 4, la boucle intérieure a 0 passage.
Il n'y a donc pas d'appel de la fonction *print*.

$4+3+2+1+0=10$, donc **il y a au total 10 appels à la fonction *print***.

3

3.1

Pour chacun des deux passages dans la boucle extérieure, le nombre d'addition est 1+ (le nombre d'additions dus à la boucle intérieure).
Il y a trois passages à chaque fois dans la boucle intérieure, comportant chacun une addition.

Le nombre total d'addition est :
$2\times(1+3\times 1)=8$
Le coût est constant, il ne dépend pas de n.

3.2

Pour chacun des deux passages dans la boucle extérieure, le nombre d'addition est 1+ (le nombre d'additions dus à la boucle intérieure).
Il y a n passages à chaque fois dans la boucle intérieure, comportant chacun une addition.

Le nombre total d'addition est :
$2\times(1+n\times 1)=2+2\cdot n$
Le coût est linéaire.

3.3

Pour chacun des n passages dans la boucle extérieure, le nombre d'addition est 1+ (le nombre d'additions dus à la boucle intérieure).
Il y a n passages à chaque fois dans la boucle intérieure, comportant chacun une addition.

Le nombre total d'addition est :
$n \times (1 + n \times 1) = n^2 + n$
Le coût est quadratique.

4

Initialisation.

Avant le passage dans la boucle, $q = 0$ et $r = a$
i.e. $b \cdot q + r = b \cdot 0 + a = a$

La propriété $a = b \cdot q + r$ est bien vérifiée.

Hérédité.

On suppose que la propriété $a = b \cdot q + r$ est vérifiée après i passages dans la boucle.
Notons q' et r' les nouvelles valeurs de q et r après le passage suivant dans la boucle.
Il faut donc montrer que $a = b \cdot q' + r'$ est bien vérifiée.
$q' = q + 1$ et $r' = r - b$
i.e. $b \cdot q' + r' = b \cdot (q + 1) + (r - b) = b \cdot q + b + r - b = b \cdot q + r$
Or, on sait que $a = b \cdot q + r$, donc $b \cdot q' + r' = a$.

La propriété est bien vérifiée après le passage suivant dans la boucle.

Conclusion.

On a démontré que la propriété $a = b \cdot q + r$ est un invariant de la boucle.

5

On choisit r comme variant.
La première valeur de r vaut a, et ensuite la suite des valeurs est décroissante (car b est positif) et vaut $a, a - b, a - 2b, a - 3b, \ldots$.
cette suite s'arrête au bout de n étapes, lorsque $a - n \cdot b < b$

Remarque : il est possible de s'arrêter là, car la terminaison de l'algorithme a été montrée. On peut cependant aller plus loin et calculer le nombre n d'étapes.

$a - n \cdot b < b$ i.e. $a < (n + 1) \cdot b$ i.e. $a < n \cdot b + b$ i.e. $a - b < n \cdot b$
i.e. $\frac{a-b}{b} < n$ car b est strictement positif.