

IHDCB331 - Algorithmique II

Devoir 3

Cédric Evrard

October 26, 2019

1 Question 1

1. Tous les éléments de $a[i]$, $a[i + 1]$, ... $a[j]$ sont égaux à zéro

```
/*@ invariant (\forallall int i;  
@             i >= 0 && i < j;  
@             a[i] == 0);  
@*/
```

2. Tous les éléments de a sont distincts

```
/*@ invariant (\forallall int i;  
@             i >= 0 && i < arrayDistinct.length;  
@             (\forallall int j; j >= 0 && j < a && i != j; a[j] !=  
@             a[i]));  
@*/
```

3. Tous les nombres de a sont pairs

```
/*@ invariant (\forallall int i;  
@             i >= 0 && i < a;  
@             a[i] % 2 == 0);  
@*/
```

4. Tous les nombres de a sont inférieurs ou égaux à 2

```
/*@ invariant (\forallall int i;  
@             i >= 0 && i < a.length;  
@             a[i] <= 2);  
@*/
```

5. Tous les nombres pairs de $a[i]$, $a[i + 1]$, ... $a[j]$ sont inférieurs à 10

```
/*@ invariant (\forallall int i;  
@             i >= 0 && i < j;  
@             (a[i] % 2 == 0 && a[i] < 10) || a[i] % 2 != 0);  
@*/
```

6. Il existe une valeur zéro dans $a[i]$, $a[i + 1]$, ... $a[j]$

```
/*@ invariant (\exists int i;  
@             i >= 0 && i < j;  
@             a[i] == 0);  
@*/
```

7. Les éléments de a sont triés par ordre croissant

```
/*@ invariant (\forall int i;  
  @           i >= 0 && i < a - 1;  
  @           a[i] <= a[i + 1]);  
@*/
```

8. x est le minium de a

2 Question 2

1. **double** max(**double** x, **double** y)

```
/*@ public normal_behavior;  
  @ assignable \nothing;  
  @ ensures \result == (x > y ? x : y);  
@*/
```

2. **boolean** contient(**double** [] a, **double** x)

```
/*@ public normal_behavior;  
  @ assignable \nothing;  
  @ ensures \result == (\exists int i;  
    @           i >= 0 && i < a.length;  
    @           x == a[i]);  
@*/
```

3. **double** max(**double** [] a)

```
/*@ public normal_behavior;  
  @ assignable \nothing;  
  @ ensures \forall (int i; i >= 0 && i < a.length; a[i] <= \result);  
  @ ensures \exists (int i; i >= 0 && i < a.length; a[i] == \result);  
@*/
```

4. **int** factorielle(**int** n)

```
/*@ public normal_behavior;  
  @ assignable \nothing;  
  @ requires n >= 1;  
  @ ensures \result == (\product int i; i >= 1 && i <= n; i);  
@*/
```

5. **int** intSqrt(**int** n)

```
/*@ public normal_behavior;  
  @ assignable \nothing;  
  @ requires n >= 0;  
  @ ensures \result * \result <= n;  
  @ ensures (\result + 1) * (\result + 1) > n;  
  @ ensures \result >= 0  
@*/
```

3 Question 10

La question 10 se base sur la signature de la méthode suivante : *public boolean enLigne(int[] a, int n, int x)* où le tableau de **String** à été remplacé par un tableau de **int**

```
/*@ public normal_behavior;
   @ assignable \nothing;
   @ ensures \result == (\exists int i; i >= 0 && i < a.length - n;
   @       (\forall int j; j >= i && j <= i + n; a[j] == x));
   @*/
```

4 Question 12

```
/*@ public normal_behavior;
   @ assignable \nothing;
   @ ensures \result == (\sum int i; i >= 0 && i < l; a[i]);
   @*/
public boolean int somme(int[] a, int l) {
    int somme = 0;
    int i = 0;

    /*@ loop_invariant i >= 0 && i < l;
       @ loop_invariant somme == (\sum int j; i >= 0 && j < i; a[j]);
       @ decreases l - i;
       @*/
    while (i < l) {
        somme = somme + a[i];
        i = i + 1;
    }

    return somme;
}
```

5 Question 13

a

6 Question 15

b

7 Question 22

c