Médian – LO42
solution

Aucun document autorisé (la copie ou les idées du voisin non plus). Le barème est indicatif.
Le soin donné à la rédaction sera évalué. Toute réponse devra être claire et justifiée (toute ambiguïté sera mal interprétée). L'élégance de la solution sera jugée.
Sauf indication contraire, dans le cas d'algorithmes les réponses doivent être rédigées en pseudo code.

Vous devez rendre les feuilles 2 et 3 complétées.

1 Faites vos preuves (6)

Calcul du PGCD

Vous disposez d'un algorithme du PGCD basé sur les propriétés arithmétiques suivantes :
-si a=b, PGCD(a,b)=a=b;
-si a<b, PGCD(a,b)=PGCD(a,b-a)
-si a>b, PGCD(a,b)=PGCD(a-b,b)

```
Var A, B, a, b : entiers ;
a ← A ;
b ← B ;
Tantque a<>b Faire
        Si a<b Alors b ← (b - a)
        Sinon a ← (a – b)
        Finsi ;
Ftq
```

On a fait la preuve de correction partielle de l'algorithme. Une malencontreuse panne d'encre nous a privés de quelques morceaux ou explications. Apportez votre science pour porter secours à de pauvres enseignants en détresse, en complétant de façon adéquate les horribles blancs disgracieux dans cette belle démonstration (feuilles 2 et 3).

2 La récursivité, effacer, vous pourrez ! (6)

Donnez les schémas de récursion de type 1 et 2 ainsi que leurs équivalents itératifs.
Y a-t-il des contraintes d'utilisation de ces transpositions ? Si oui expliquez pourquoi.

3 La sélection non naturelle (des plus faibles) (7)

On souhaite connaître le nombre d'éléments minimaux et l'indice de la première occurrence d'un élément minimal dans un tableau de N entiers (on suppose N >= 2).
Exemple: si la suite des éléments du tableau est :

5,2,4,2,1,7,9,4,1,1

il y a 3 occurrences de l'élément minimal et l'indice de sa première occurrence est 5.

a) Ecrire un algorithme itératif donnant le résultat souhaité. On précisera la nature de l'algorithme (fonction ou procédure), ainsi que le type et les modes de passage des paramètres.

b) Evaluer la complexité de votre algorithme.

c) Prouver que, si on appelle T le tableau, NMIN le nombre d'occurrences de l'élément minimal et RMIN l'indice de sa première occurrence, on a l'invariant de boucle :

$$(\forall j \in [1..\ RMIN[,\ T[j]>T[RMIN],\ \forall k \in [RMIN..\ i[,\ T[k] \geq T[RMIN]) \wedge$$
$$NMIN = \left| \{k \text{ tels que } k \in [1..i-1] \wedge T[k]=T[RMIN] \} \right|$$

Solution :

a) La lecture de l'énoncé avec le point c devrait vous amener à élaborer un algorithme avec une seule boucle.

L'algorithme doit fournir deux résultats (le rang et le nombre d'éléments) donc c'est une procédure. Le tableau T sera passé par référence pour éviter sa recopie au niveau de la procédure qui coûte N additions si le tableau est de longueur N. La longueur du tableau sera passée par valeur. Les paramètres rang (RMIN) et nombre d'occurrence (NMIN) seront passés par référence pour que la procédure puisse les modifier.

```
procédure mini(var T: tableau [1..N] d'entier , var RMIN,NMIN: entier);
var i: entier;
début
    RMIN←1;
    NMIN←1;
    pour i de 2 à N faire
        si T[i]<T[RMIN] alors
            RMIN←i;
            NMIN←1
        sinon
            si T[i]=T[RMIN] alors NMIN←NMIN+1
          fsi
        fsi
    finpour
fin
```

b) Il y a une boucle pour de 2 à N. Donc la complexité est N.
Dans cet algorithme il n'y a pas d'arrêt de boucle avant N donc pas de pire et moyen cas.

c) Pour montrer que c'est un invariant il faut vérifier que la condition est vraie avant la boucle, qu'elle est vraie après la boucle et que le passage dans la boucle laisse la condition vraie.

A l'entrée de la boucle on a i=2. RMIN=1, NMIN=1 et T[1..i-1] est donc réduit à T[1], or RMIN=1, NMIN=1 donc l'assertion est vérifiée.
Supposons que l'assertion est vérifiée au rang i, vérifions-la après un tour supplémentaire de la boucle. Il y a trois cas de figure :
si T[i]<T[RMIN] alors T[i] est le plus petit des éléments de T[1..i], et c'est sa première occurrence. On a RMIN=i et NMIN=1 donc après incrémentation de i, l'assertion est vérifiée.
si T[i]=T[RMIN] alors par hypothèse, dans T[1..i-1] il y a NMIN occurrence de T[RMIN]. Il y a donc NMIN+1 occurrences de T[RMIN] dans T[1..i]. On incrémente NIMN et i de 1 donc après ces opérations l'assertion est vraie.
sinon il y a n'y a pas de nouvelle occurrence de T[RMIN] donc il y a NMIN occurrence de T[RMIN] dans T[1..I]. Donc après incrémentation de I, l'assertion est vérifiée.
En sortie de boucle on a i =N+1 et on a bien NMIN égal au nombre d'occurrences de T[RMIN] dans T[1..N].

Démonstration plus formelle

Avant la boucle : RMIN=1, NMIN=1 et i=2 c'est l'initialisation. Donc

$\forall j \in [1..\ 1[,\ T[j]>T[RMIN]$,	l'intervalle est vide donc l'expression est vraie
$\forall k \in [RMIN..\ i[,\ T[k] \geq T[RMIN]$	L'intervalle est limité à T[RMIN] donc vraie
$NMIN= \lvert \{k \text{ tels que } k \in [1..i-1] \wedge T[k]=T[RMIN]\} \rvert$	Une seule valeur possible pour k=1 donc vraie

Après la boucle : i =N+1

$\forall j \in [1..\ RMIN[,\ T[j]>T[RMIN]$,	Ce sont les conditions recherchées
$\forall k \in [RMIN..\ N+1[,\ T[k] \geq T[RMIN]$	
$NMIN= \lvert \{k \text{ tels que } k \in [1..N] \wedge T[k]=T[RMIN]\} \rvert$	

Le corps de la boucle :

La boucle se termine par une incrémentation du compteur :

$(\forall j \in [1.. RMIN[, T[j]>T[RMIN], \forall k\in [RMIN.. i+1[, T[k]\geq T[RMIN]) \wedge$
$NMIN= | \{k \text{ tels que } k\in [1..i] \wedge T[k]=T[RMIN] \} |$

Finpour $\equiv i \leftarrow i + 1$;

Le corps du Alors

$(\forall j \in [1.. i[, T[j]>T[i], \forall k\in [i.. i+1[, T[k]\geq T[i]) \wedge$
$1= | \{k \text{ tels que } k\in [1..i] \wedge T[k]=T[i] \} |$

RMIN←i;
NMIN←1

$(\forall j \in [1.. RMIN[, T[j]>T[RMIN], \forall k\in [RMIN.. i+1[, T[k]\geq T[RMIN]) \wedge$
$NMIN= | \{k \text{ tels que } k\in [1..i] \wedge T[k]=T[RMIN] \} |$

Or :

$T[i]<T[RMIN]$
$(\forall j \in [1.. RMIN[, T[j]>T[RMIN], \forall k\in [RMIN.. i [, T[k]\geq T[RMIN]) \wedge$
$NMIN= | \{k \text{ tels que } k\in [1..i[\wedge T[k]=T[RMIN] \} |$
$\Rightarrow \quad (\forall j \in [1.. i[, T[j]>T[i], \forall k\in [i.. i+1[, T[k]\geq T[i]) \wedge$
$1= | \{k \text{ tels que } k\in [1..i] \wedge T[k]=T[i] \} |$

Par renforcement :

$T[i]<T[RMIN]$
$(\forall j \in [1.. RMIN[, T[j]>T[RMIN], \forall k\in [RMIN.. i [, T[k]\geq T[RMIN]) \wedge$
$NMIN= | \{k \text{ tels que } k\in [1..i[\wedge T[k]=T[RMIN] \} |$

RMIN←i;
NMIN←1

Le corps du sinon :

Le corps du alors (imbriqué)

si T[i]=T[RMIN] alors

$(\forall j \in [1.. RMIN[, T[j]>T[RMIN], \forall k\in [RMIN.. i+1[, T[k]\geq T[RMIN]) \wedge$
$NMIN+1= | \{k \text{ tels que } k\in [1..i] \wedge T[k]=T[RMIN] \} |$

{ NMIN←NMIN+1 }

$(\forall j \in [1.. RMIN[, T[j]>T[RMIN], \forall k\in [RMIN.. i+1[, T[k]\geq T[RMIN]) \wedge$
$NMIN= | \{k \text{ tels que } k\in [1..i] \wedge T[k]=T[RMIN] \} |$

Par la règle de renforcement

$T[i]=T[RMIN] \wedge (\forall j \in [1.. RMIN[, T[j]>T[RMIN],$
$\forall k\in [RMIN.. i [, T[k]\geq T[RMIN]) \wedge$
$NMIN= | \{k \text{ tels que } k\in [1..i-1] \wedge T[k]=T[RMIN] \} |$

{ NMIN←NMIN+1 }

$(\forall j \in [1.. RMIN[, T[j]>T[RMIN], \forall k\in [RMIN.. i+1[, T[k]\geq T[RMIN]) \wedge$
$NMIN= | \{k \text{ tels que } k\in [1..i] \wedge T[k]=T[RMIN] \} |$

Pour le sinon on peut renforcer avec $T[i]>T[RMIN]$ donc on peut écrire

$T[i]>T[RMIN] \wedge (\forall j \in [1.. RMIN[, T[j]>T[RMIN],$
$\forall k\in [RMIN.. i [, T[k]\geq T[RMIN]) \wedge$
$NMIN= | \{k \text{ tels que } k\in [1..i-1] \wedge T[k]=T[RMIN] \} |$

Par la règle du si on obtient

$T[i]\geq T[RMIN] \wedge (\forall j \in [1.. RMIN[, T[j]>T[RMIN],$
$\forall k\in [RMIN.. i [, T[k]\geq T[RMIN]) \wedge$
$NMIN = | \{k \text{ tels que } k\in [1..i-1] \wedge T[k]=T[RMIN] \} |$

si T[i]=T[RMIN] alors NMIN←NMIN+1 Fsi

$(\forall j \in [1.. RMIN[, T[j]>T[RMIN], \forall k\in [RMIN.. i+1[, T[k]\geq T[RMIN]) \wedge$
$NMIN= | \{k \text{ tels que } k\in [1..i] \wedge T[k]=T[RMIN] \} |$

L'instruction si : par la règle de la conditionnelle

$(\forall j \in [1.. RMIN[, T[j]>T[RMIN], \forall k\in [RMIN.. i [, T[k]\geq T[RMIN]) \wedge$
$NMIN= | \{k \text{ tels que } k\in [1..i-1] \wedge T[k]=T[RMIN] \} |$

Si T[i]<T[RMIN] alors … Fsi

$(\forall j \in [1.. RMIN[, T[j]>T[RMIN], \forall k\in [RMIN.. i+1[, T[k]\geq T[RMIN]) \wedge$

$$NMIN = \left| \{k \text{ tels que } k \in [1..i] \wedge T[k]=T[RMIN]\} \right|$$

On retrouve bien la condition donc c'est un invariant.

$(\forall j \in [1..\ RMIN[,\ T[j]>T[RMIN],\ \forall k \in [RMIN..\ i[,\ T[k] \geq T[RMIN]) \wedge$
$NMIN = \left| \{k \text{ tels que } k \in [1..i\text{-}1] \wedge T[k]=T[RMIN]\} \right| \wedge i=N+1$

1) Le PGCD

Démonstration :

Prenons donc pour invariant de boucle I : pgcd(A;B) = pgcd(a;b) $\wedge$ A, B, a, b > 0, et voyons qu'il est effectivement préservé par la boucle.

Corps du *alors* Par l'axiome d'affectation on a :

$$\begin{pmatrix} A, B, a, (b-a) > 0 \\ pgcd(A,B) = pgcd(a, b-a) \end{pmatrix} \{b \leftarrow b-a\} \begin{pmatrix} A, B, a, b > 0 \\ pgcd(A,B) = pgcd(a,b) \end{pmatrix}$$

D'après les propriétés du pgcd on a (a <> b est contenu dans b > a, mais cette redondance permet de faire apparaître la condition du *tant que*) :

$$\begin{pmatrix} b > a \\ a \neq b \\ A, B, a, b > 0 \\ pgcd(A,B) = pgcd(a,b) \end{pmatrix} \Rightarrow \begin{pmatrix} A, B, a, (b-a) > 0 \\ pgcd(A,B) = pgcd(a, b-a) \end{pmatrix}$$

Par la règle de renforcement de la pré-condition on a :

$$\begin{pmatrix} b > a \\ a \neq b \\ A, B, a, b > 0 \\ pgcd(A,B) = pgcd(a,b) \end{pmatrix} \{b \leftarrow b-a\} \begin{pmatrix} A, B, a, b > 0 \\ pgcd(A,B) = pgcd(a,b) \end{pmatrix}$$

C'est-à-dire

$$\begin{pmatrix} b > a \\ a \neq b \\ I \end{pmatrix} \{b \leftarrow b-a\} \ (I) \qquad (*)$$

Corps du *sinon* Par l'axiome d'affectation on a :

$$\begin{pmatrix} A, B, (a-b), b > 0 \\ pgcd(A,B) = pgcd(a-b, b) \end{pmatrix} \{a \leftarrow a-b\} \begin{pmatrix} A, B, a, b > 0 \\ pgcd(A,B) = pgcd(a,b) \end{pmatrix}$$

D'après les propriétés du pgcd on a (a ≠ b et a ≥ b équivaut à a > b, mais on souhaite faire apparaître la négation de a < b, a = b : c'est la condition du *si* qui permet d'appliquer la règle du *si*, et cela fait aussi apparaître la condition du *tant que* a ≠ b)

$$\begin{pmatrix} a \geq b \\ a \neq b \\ A, B, a, b > 0 \\ \mathrm{pgcd}(A,B) = \mathrm{pgcd}(a,b) \end{pmatrix} \Rightarrow \begin{pmatrix} A, B, (a - b), b > 0 \\ \mathrm{pgcd}(A,B) = \mathrm{pgcd}(a - b, b) \end{pmatrix}$$

Par la règle de renforcement de la pré-condition on a (en prenant X = pgcd(A, B)) :

$$\begin{pmatrix} a \geq b \\ a \neq b \\ A, B, a, b > 0 \\ \mathrm{pgcd}(A,B) = \mathrm{pgcd}(a,b) \end{pmatrix} \{a \leftarrow a - b\} \begin{pmatrix} A, B, a, b > 0 \\ \mathrm{pgcd}(A,B) = \mathrm{pgcd}(a,b) \end{pmatrix}$$

C'est-à-dire

$$\begin{pmatrix} a \geq b \\ a \neq b \\ I \end{pmatrix} \{a \leftarrow a - b\} \ (I) \qquad (**)$$

Le *si alors sinon finsi* D'après (*) et (**) par la règle du *si alors sinon finsi* on a :

$$\begin{pmatrix} a \neq b \\ I \end{pmatrix} \{\mathrm{si}\ a < b\ \mathrm{alors}\ ...\ \mathrm{finsi}\} \ (I) \qquad (***)$$

Le *tant que a<>b faire. . . fintantque* D'après la règle du *tant que* on déduit de (***) :

$$(I) \ \{\mathrm{tantque}\ a <> b\ \mathrm{faire} ... \mathrm{fintantque}\} \ (I \wedge a = b)$$

Or on sait que si a > 0 alors pgcd(a, a) est défini et vaut a donc :

$$I \wedge a = b \rightarrow \mathrm{pgcd}(A, B) = a = b$$

En appliquant la règle de la post-condition à (****) on obtient :

$$(I) \ \{\mathrm{tantque}\ a <> b\ \mathrm{faire} ... \mathrm{fintantque}\} \ (\mathrm{pgcd}(A,B) = a = b) \qquad (\#)$$

Affectations initiales On a (en omettant la répétition A > 0 et A > 0) l'axiome d'affectation suivant :

$$\begin{pmatrix} A, B > 0 \\ \mathrm{pgcd}(A,B) = \mathrm{pgcd}(A,B) \end{pmatrix} \{a \leftarrow A\} \begin{pmatrix} A, B, a, > 0 \\ \mathrm{pgcd}(a,B) = \mathrm{pgcd}(A,B) \end{pmatrix}$$

et

$$\begin{pmatrix} A, B, a > 0 \\ \mathrm{pgcd}(a,B) = \mathrm{pgcd}(A,B) \end{pmatrix} \{b \leftarrow B\} \begin{pmatrix} A, B, a, b > 0 \\ \mathrm{pgcd}(a,b) = \mathrm{pgcd}(A,B) \end{pmatrix}$$

Par composition séquentielle entre ces deux axiomes d'affectation (la post-condition du premier est bien la pré-condition du second) on obtient :

$$\begin{pmatrix} A, B > 0 \\ pgcd(A, B) = pgcd(A, B) \end{pmatrix} \ \{a \leftarrow A ; b \leftarrow B\} \ \begin{pmatrix} A, B, a, b > 0 \\ pgcd(a, b) = pgcd(A, B) \end{pmatrix}$$

et comme A, B > 0 $\rightarrow$ (A, B > 0 $\wedge$ pgcd(A, B) = pgcd(A, B)) est toujours vraie on a par règle de la pré-condition :

$A, B > 0 \ \{ a \leftarrow A ; b \leftarrow B \} \ pgcd(a, b) = pgcd(A, B)$ (##)

Spécification de tout le programme Par composition séquentielle sur (#) et (##), on a donc, pour le programme tout entier — et par règle de la post-condition , on peut ne garder qu'une partie de celle-ci si on le souhaite :

$A, B > 0 \ \{ p \} \ (A, B, a, b > 0 \wedge pgcd(a, b) = a = b)$