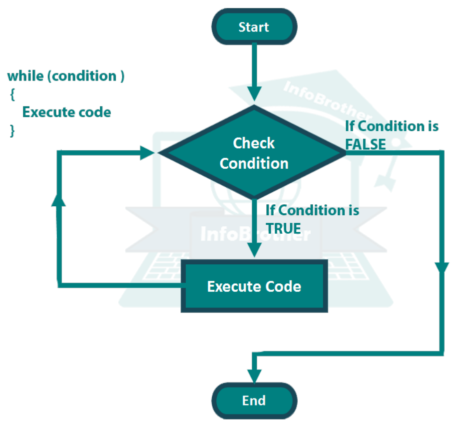
TD Terminaison d'algorithme

Olivier Raynaud (raynaud@isima.fr)



Exercice 1. *Etudier les tests d'arrêt des algorithmes 1, 2, 3.*

Algorithme 1: $factorielle()$

Données : N : entier ; $(N > 0)$

Résultat : f : entier ; la valeur de la fonction factorielle pour N ;

début

  $n \leftarrow N$; $f \leftarrow fact(7)$;

  tant que $(n \neq 7)$ **faire**

    si $(n > 7)$ **alors**

      $f \leftarrow f * n$; $n \leftarrow n - 1$;

    sinon

      $n \leftarrow n + 1$; $f \leftarrow f/n$

  retourner f ;

fin

Algorithme 2: *diverge*()

Données : N : entier ; $(N \geq 0)$

Résultat : n : entier ;

début

$n \leftarrow N$;

tant que $(n \neq 0)$ **faire**

si *(n est pair)* **alors**

$n \leftarrow 2 * n$;

sinon

$n \leftarrow n - 1$

retourner n ;

fin

Algorithme 3: *converge*()

Données : N : entier ; $(N > 0)$

Résultat : n : entier ;

début

$n \leftarrow N$;

tant que *($n \neq 1$)* **faire**

si *(n est pair)* **alors**

$n \leftarrow n/2$;

sinon

$n \leftarrow n + 1$;

retourner n ;

fin

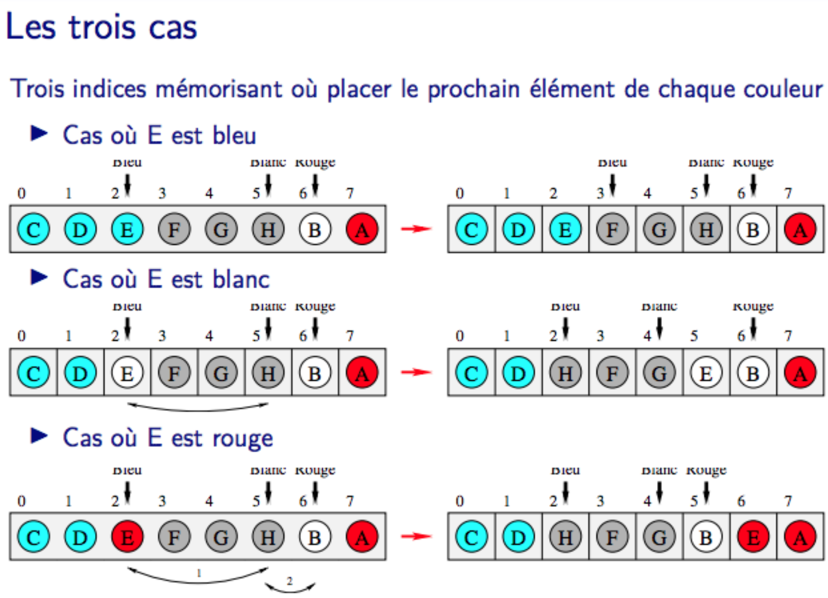
Exercice 2 (Le drapeau Hollandais).
Considérons le problème suivant :

Problème 1 (Le drapeau Hollandais).
Entrée : $tab[1..n]$ *: tableau de couleurs (bleu, blanc et rouge) ;*
Sortie : $tab[1..n]$ *: le tableau* $tab[]$ *est rangé, c'est à dire que toutes les couleurs bleues sont à gauche, les rouges à droite et les blanches au centre.*

Algorithme 4: *drapeauHollandais*()

```
Données : tab[1..n] : tableau de couleurs;
Résultat : tab[1..n] : tableau de couleurs;
Variables : indiceBleu, indiceRouge, compteur : entier;
début
    indiceBleu ← 1; indiceRouge ← N;
    compteur ← 1;
    tant que (compteur ≤ indiceRouge) faire
        si (tab[compteur] = bleu) alors
            tab[] ← echange(tab[], compteur, indiceBleu);
            indiceBleu ← indiceBleu + 1;
            compteur ← compteur + 1;
        sinon
            si (tab[compteur] = rouge) alors
                tab[] ← echange(tab[], compteur, indiceRouge);
                indiceRouge ← indiceRouge − 1;
            sinon
                compteur ← compteur + 1;
    retourner tab[];
fin
```

Question 1. *L'Algorithme 4 répond au Problème 1. Donner la complexité de l'algorithme et montrer sa terminaison.*



(Note :image extrait de https ://algo.gricad-pages.univ-grenoble-alpes.fr/L3I-S5-algo/Cours03.pdf)

Exercice 3. *Montrer la terminaison l'algorithme du calcul du PGCD d'Euclide (Cf. algorithme 5).*

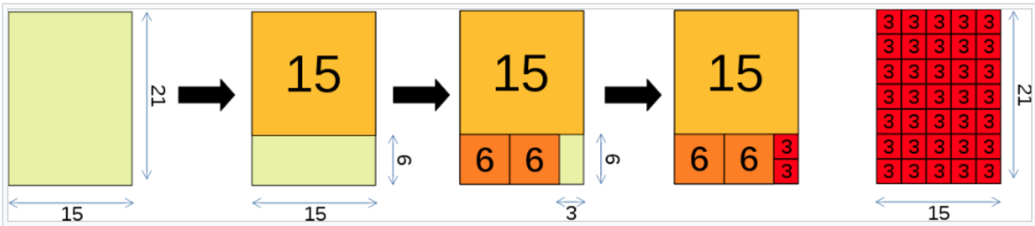


FIGURE 1 – Interprétation géométrique du principe de l'algorithme d'Euclide (wikipedia)

Algorithme 5: *Euclide*(N, M)

Données : N, M : entier ;
Résultat : : entier (le plus grand commun diviseur des entiers N et M) ;
Variables : n et m : entier ;
début
  $n \leftarrow N$; $m \leftarrow M$;
  tant que $(n \neq 0)$ **faire**
    si $(m > n)$ **alors**
      $m \leftarrow m - n$;
    sinon
      $n \leftarrow n - m$
  retourner m ;
fin

Exercice 4 (Entiers signés dans un tableau).

Soit un tableau d'entiers signés de dimension $n \times m$.

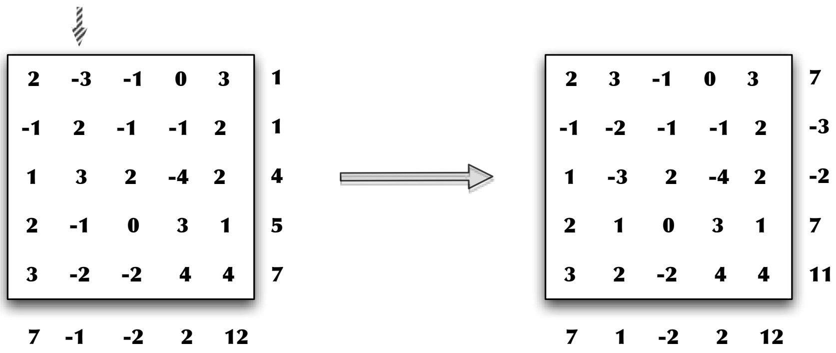


FIGURE 2 – Matrice d'entiers dont les signes de la seconde colonne ont été inversés.

Question 1. *Soit l'opération consistant à changer simultanément tous les signes des entiers d'une même ligne ou d'une même colonne. Montrer la terminaison de l'algorithme 6.*

Algorithme 6: Inversion

Données : $T[]$: tableau d'entiers ;

Résultat : Néant, Toutes les lignes et les colonnes du tableau admettent des sommes non négative.

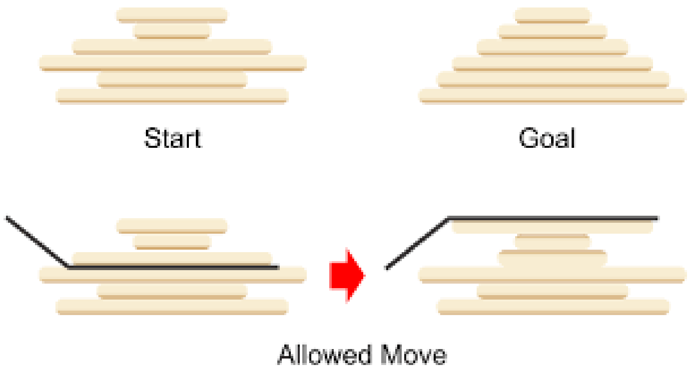
début

  tant que *(Il existe une ligne ou une colonne de somme négative dans $T[]$)* **faire**

    Inverser les signes de la colonne ou de la ligne ;

fin

Exercice 5 (Problème des pancakes).



Soit un tableau $t[]$ de taille n contenant une permutation de l'ensemble $[1, .., n]$. Soit la fonction $inverser(t[], i, j)$ qui retourne le tableau $t[]$ dans lequel toutes les valeurs entre les indices i et j ont »t» inversées.

Exemple : $t[] = \{1, 4, 7, 3, 5, 2, 6\}$ *;* $inverser(t[], 2, 5)$ *retourne* $t[] = \{1, 5, 3, 7, 4, 2, 6\}$.

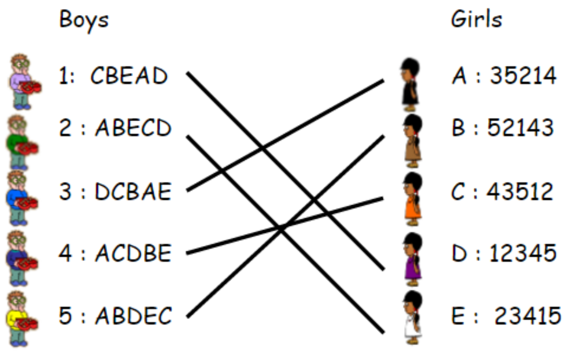
Question 1. *Montrer la terminaison de l'algorithme 7.*

Algorithme 7: $perturbation()$

Données : $t[]$: tableau d'entiers ;
Résultat : :
début
  tant que $(t[1] \neq 1)$ **faire**
    $t[] \leftarrow inverser(t[], 1, t[1])$;
  retourner $t[]$;
fin

Exercice 6 (**Mariages stables**).

Soient n hommes et n femmes chacun ayant des préférences strictes sur les personnes du sexe opposé. On appelle mariage un appariement de chaque homme avec une et une seule femme. Un couple (H, F) non apparié est dit **instable** *si H (resp. F) préfère F (resp. H) à son conjoint officiel. Un mariage est dit* **stable** *s'il ne contient aucun couple instable.*



Gale et Shapley (1962) démontrent par construction l'existence d'un mariage stable, quelque soit l'instance. L'algorithme est le suivant :

Algorithme 8: *Gale et Shapley*(1962)

Données : Listes de préférence de chaque homme et de chaque femme ;
Résultat : Mariage stable
début
    tant que *(Il existe un homme libre n'ayant pas courtisé chaque femme)* **faire**
        Choisir m l'un d'entre eux ;
        Soit w la première femme non courtisée de m ;
        si *(w est libre)* **alors**
            Apparier m et w ;
        sinon
            si *(w préfère m et son partenaire courant m')* **alors**
                Apparier m et w ;
                Libérer m' ;
            sinon
                w rejette m ;
fin

Question 1. *Montrer la terminaison de l'algorithme 8.*

Question 2. *Montrer que l'Algorithme 8 produit un mariage stable.*