

Errata (13 avril 2017)

Voici une liste de coquilles et précisions correspondant au livre **Option Informatique MPSI-MP/MP*** par Roger Mansuy, Vuibert 2014. Merci aux lecteurs qui ont signalé ces erreurs. N'hésitez pas à signaler d'autres erreurs ou imprécisions à l'adresse électronique

roger.mansuy@gmail.com

Chapitre 4

Dans la fonction auxiliaire page 71, le cas de la retenue non nulle est oublié en ligne 2. Voici le code corrigé :

```
let rec aux p1 p2 r resultat =
  if (est_vide p1) && (est_vide p2) && r <> 0 then
    (Empile r resultat)
  else if (est_vide p1) then begin
      (Empile 0 p1);
      aux p1 p2 r resultat
    end
  else if (est_vide p2) then aux p2 p1 r resultat
  else begin
      let a = depile p1 and b = depile p2 in
      let r' = (a+b+r)/10 and q = (a+b+r) mod 10 in
      (Empile q resultat);
      aux p1 p2 r' resultat;
    end;;
```

Par ailleurs, dans l'implémentation des files pages 72 et 73, les mots début et fin ont été inversés. Voici la version corrigée

```
type 'a file = {mutable debut:'a list; mutable fin:'a list};;

let File_vide () = {debut=[]; fin=[]};;

let Ajoute e f = f.debut <- e::f.debut; f;;

let est_vide f = (f.debut=[])&&(f.fin=[]);;

let premier f =
```

```

if (est_vide f) then failwith "file vide"
else begin
  if (f.fin=[]) then begin
    f.fin <- rev f.debut;
    f.debut <- []
  end;

  hd f.fin
end;;

let queue f =
  if (est_vide f) then failwith "file vide"
  else begin
    if (f.fin=[]) then begin
      f.fin <- rev f.debut;
      f.debut <- []
    end;

    f.fin <- tl f.fin;
    f
  end;;

```

La même inattention est présente dans l'énoncé de l'exercice 13 question 2.

Chapitre 9

Dans la fonction de calcul naïf des plus proches voisins dans un nuage de points page 185, il y a une erreur d'indice dans la boucle sur j. Voici le code corrigé :

```

let distance a b =
  sqrt ((fst(a)-.fst(b))**2. +. (snd(a)-.snd(b))**2.);;

let plusproche t =
  let n = vect_length t in
  let a = ref 0 and b = ref 1 in
  let d = ref (distance t.(!a) t.(!b)) in
  for i = 0 to (n-1) do
    for j = i+1 to (n-1) do
      let d' = (distance t.(i) t.(j)) in
      if d' < !d then
        begin
          d:=d';
          a:=i;
          b:=j
        end;
    end;
  end;

```

```
done;  
done;  
t.(!a),t.(!b) ;;
```

Dans la seconde remarque page 187, une erreur s'est glissée dans la complexité annoncée $n^2 \log n$ au lieu de $n(\log n)^2$. Voici la remarque corrigée :

Remarque

Si on décide de ne pas trier initialement le nuage de points mais de ne trier que les points de la bande T au moment de l'étape de « fusion », alors le coût de cette fusion devient quasilinear en le nombre de points de T et, dans le pire des cas $\Theta(n \log n)$. Dans ce cas, la complexité dans le pire des cas v_n pour trier n points vérifie une équation de la forme $v_n = v_{\lfloor \frac{n}{2} \rfloor} + v_{\lfloor \frac{n}{2} \rfloor} + \Theta(n \log n)$ donc $v_n = \Theta(n(\log n)^2)$. Cet algorithme est moins performant dans le pire des cas.