

Solutions de la partie théorique

1. a) Algorithme 3.2.1 du cours.
- b) Soit i la valeur du compteur de la boucle extérieure. Le nombre de comparaisons de strings (ligne 11 de l'algorithme) est indiqué dans la dernière colonne du tableau.

i	liste	n_C
	P Y T H O N	
1	P Y T H O N	1
2	P T Y H O N	2
3	H P T Y O N	3
4	H O P T Y N	4
5	H N O P T Y	5

- c) D'après la dernière colonne du tableau précédent, on a $1 + 2 + 3 + 4 + 5 = 15$ comparaisons de strings au total.
- d) Lors du tri de n strings, la variable i prend successivement les valeurs $1, 2, \dots, n-1$.
 Nombre minimal de comparaisons (p. ex. pour une liste déjà triée) : une seule pour chaque valeur prise par i . Au total : $n - 1$ comparaisons.
 Nombre maximal de comparaisons (p. ex. pour une liste initialement triée dans l'ordre inverse) : i comparaisons pour chaque valeur prise par i , c'est-à-dire au total

$$\sum_{i=1}^{n-1} i = \frac{(n-1)n}{2} = \frac{n^2 - n}{2} \text{ comparaisons.}$$

2. a) **function** `f(x:extended):extended;`
begin
`f:=abs(x)-sqrt(sqr(sqr(x))+1)`
end;

Algorithme 5.1 du cours, mais avec couleur bleue (remplacer `clRed` par `clBlue` à la ligne 14).

- b) Ajouter entre les lignes 4 et 5 de l'algorithme 5.1 le code suivant, qui fait tracer un rectangle blanc recouvrant tout le canevas :

```
im.Canvas.Pen.Color:=clWhite;
im.Canvas.Brush.Color:=clWhite;
im.Canvas.Rectangle(0,0,im.Width,im.Height);
```

- c) Ajouter entre les lignes 13 et 14 de l'algorithme 5.1 le code suivant :

```
im.Canvas.Pen.Color:=clGreen;
if (x1<=0) and (x2>=0) then begin { axe des ordonnées visible }
  im.Canvas.MoveTo(round(-x1/dx),0);
  im.Canvas.LineTo(round(-x1/dx),im.Height)
end;
if (ymin<=0) and (ymax>=0) then begin { axe des abscisses visible }
  im.Canvas.MoveTo(0,round(ymax/dy));
  im.Canvas.LineTo(im.Width,round(ymax/dy))
end;
```