Les Algorithmes De Tri

Trie à Bull

```
0) Def Proc bull(var t :tab; n :entier)
1) Repéter
        Permut ← vrais
        Pour i de 1 à n-1 faire
              Si(t[i]<t[i+1])alors
                    Proc permuter(t[i],t[i+1])
                    Permut ← faux
              Fin si
        Fin pour
   Jusqu'à (permut)
2) Fin bull
```

Procédure Permuter

```
0) Def Proc permuter(var A,B :entier)
1) Vaux ← A
2) A ← B
3) B ← C
4) Fin permuter
```

Tri par Sélection

```
0) Def Proc selection(var t :tab; n :entier)
1) Pour i de 1 a n faire
        Pmin (ou pma) ← Fn mini(i,n)(ou Fn maxi(i,n))
        Si (i<>pmin(ou pmax pour maximum)) alors
              Proc permuter(t[i],t[pmin(ou pmax)])
        Fin si
   Fin pour
2) Fin selection
```

Fonction Mini

```
0) Def Fn mini(a,b :entier):entier
1) [pmin←a]
        pour i de a+1 à b faire
              Si(t[pmin]>t[i])alors
                    Proc permuter(t[pmin],t[i])
              Fin si
        Fin pour
2) Mini ← pmin
```

Procédure Permuter

```
0) Def proc permuter(var A,B :entier)
1) Vaux ← A
2) A ← B
3) B ← C
4) Fin permuter
```

M® MÂATOUG Mourad(Les Algorithmes de tri)

Tri par Insertion

```
0) Def Proc insertion (var t :tab; n :entier)
1) Pour i de 2 à n faire
        Vaux ← t[i]
        J ← i
        Tantque (j>1) et (t[j-1] < (ou >) vaux) faire
              t[j] ← t[j-1]
              j ← j-1
        fin tantque
   fin pour
2) Fin insertion
```

Tri Shell

```
0) Def Proc shell (var t :tab; n :entier)
1) [pas←1]Tantque (pas<n) faire
        Pas ← pas*3+1
   Fin tantque
2) Pour i de pas+1 à n faire
        Vaux ← t[i]
        J  ← i
        Tantque (j>pas) et (t[j-pas]< (ou >) vaux) faire
              T[j] ← t[j-pas]
              J ← j-p
        Fin tantque
        T[j] ← vaux
   Fin pour
3) Fin shell
```

Classement

```
0) Def Proc classement (var t :tab; n :entier)
1) Pour i de 1 à n faire
        S ← Fn compter (t,n,t[i])
        R[s+1] ← t[i]
   Fin pour
2) Fin classement
```

Fonction Compter

```
0) Def Fn classement (var t :tab; n,x :entier) :entier
1) [s←0]Pour i de 1 à n faire
        Si (t[i]< (ou >) x) alors
              s ← s+1
        Fin si
   Fin pour
2) Compter ← s
Fin compter
```

M® MATOUG Mourad (Les Algorithmes de tri)

Procédure fusion

```
0) Def Proc fusion (var t :tab; d,mil,f :entier)
1) i←d
2) j←mil+1
3) k←d
4) Tantque (i<=mil) et (j<=f) faire
        Si(t[i]< (ou>) t[j])alors
              p[k]←t[i]
              i←i+1
        Sinon
              p[k]←t[j]
              j←j+1
        fin si
        k←k+1
   fin tantque
5) Si(i>mil)alors
        Pour l de i à f faire
              p[k]←t[l]
              k←k+1
        fin pour
   Sinon
        Pour l de i à mil faire
              p[k]←t[l]
              k←k+1
        fin pour
   fin si
6) Pour l de d à f faire
        t[l]←p[l]
   fin pour
7) Fin fusion
```

Tri Fusion

```
0) Def Proc tri_fusion (var t :tab; d,f:entier)
1) Si (d<f) alors
        mil← (d+f)div 2
        Proc tri_fusion(t,d,mil)
        Proc tri_fusion(t,mil+1,f)
        Proc fusion(t,d,mil,f)
   fin si
2) Fin tri_fusion
```

M® MAAtOUG Mourad (Les Algorithmes de tri)