Tri selection

Principe : Consiste à répéter le traitement $n-1$ fois :
- rechercher la position du min
- permuter si le min n'est pas dans bon pos

Si (n = 6) alors : 5 itérations : rechercher min
↳ si pas dans bn pos
⇒ permuter

0	1	2	3	4	5
20	70	13	-7	21	-10

: itération 1 : ($i=0$) : rechercher min
↳ supposition : posmin ← [0]
↳ recherche sur tout le tableau
⇒ boucle pour
↳ condition : si

Algorithme
Début

```
pour i de 0 à n-2 faire
    posmin ← i
    pour j de i+1 à n-1 faire
        si t[j] < t[posmin] alors
            posmin ← j
        fin si
    fin pour
    si i ≠ posmin alors
        ...
    fin si
```

étape 1 : rechercher position du min

étape 2 : comparer i avec posmin (Vf si element est dans la bonne position

0	1	2	3
20	51	-1	10

posmin [0] ; [0] < [0] ✗ (j)
↳ posmin
↳ [1] < [0] ✗ (posmin, j)
↳ [2] < [0] ✓ (posmin, j) ⇒ posmin ← i

i ← 0

posmin [2] ; [3] < [2] ✗ j ← 3

⇒ posmin ≠ 2 etape 1
• etape 2 : permuter si posmin ≠ i : 2 ≠ 0 ✓
⇒ permut

---

i ← 1 ; | -1 | 51 | 20 | 10 | ; posmin = i = 1 : [51] ; [2] < posmin ?
⇒ posmin ← j ← 2

posmin [2] ; [3] < posmin ? ✓
⇒ posmin ← 3

j ← 3

⇒ posmin ← 3 : etape 1 ✓ ; etape 2 : posmin ≠ i ? ✓
⇒ permuter

0	1	2	3
-1	10	20	51

i ← 2 ; posmin ↑i [2] ; j ← i+1 = 3
[3] < [2] posmin ? : ✗

⇒ tableau trié ; Sater i ← n-2 : 2
j ← n-1 : 3