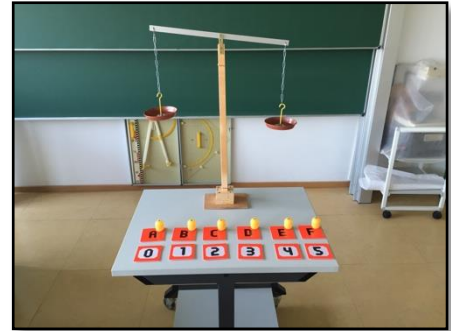


# Tri par sélection

Supposons qu'on ait une liste avec des œufs surprises de poids différents.

Comment procéder pour trier la liste des œufs par poids croissant, si on peut uniquement se servir d'une balance qui permet de comparer le poids de deux œufs ? Cette balance n'indique pas le poids en soi.



## Pseudocode

```
sortByAscendingWeight(){
```

Pour tout index  $i$  de la liste **sauf le dernier**

Trouver l'index  $iMin$  de l'œuf le plus léger **non trié**

Si  $iMin \neq i$

Echanger l'œuf à l'index  $i$  et l'œuf à l'index  $iMin$

```
}
```

## Code Java

```
public void sortByAscendingWeight() {

    for (int i = 0; i < alSurpriseEggs.size() - 1; i++) {

        SurpriseEgg eggMin = alSurpriseEggs.get(i);
        int iMin = i;

        for (int j = i+1; j < alSurpriseEggs.size(); j++) {

            SurpriseEgg egg = alSurpriseEggs.get(j);

            if (egg.getWeight() < eggMin.getWeight()) {
                eggMin = egg;
                iMin = j;
            }

        }

        if (i != iMin) {
            SurpriseEgg temp = alSurpriseEggs.get(i);
            alSurpriseEggs.set(i, eggMin);
            alSurpriseEggs.set(iMin, temp);
        }

    }

}
```

## Exemple d'exécution

Soit la liste suivante d'œufs avec les poids indiqués :

Index	0	1	2	3	4	5	6	7
Poids de l'œuf	10	8	13	3	1	15	1	8

### Première étape - Trouver l'œuf le plus léger et le placer à l'index 0

Le tri par sélection commence par chercher l'index de l'œuf le plus léger qui vaut 4 dans ce cas-ci.

Index actuel (i == 0)	Index du minimum iMin	Œufs triés		Œufs non triés				
Index	0	1	2	3	4	5	6	7
Poids de l'œuf	10	8	13	3	1	15	1	8

Puis les œufs se trouvant à l'index actuel 0 et à l'index 4 sont échangés.

Index actuel (i == 0)	Index du minimum iMin	Œufs triés		Œufs non triés				
Index	0	1	2	3	4	5	6	7
Poids de l'œuf	1	8	13	3	10	15	1	8

### Deuxième étape - Trouver l'œuf le plus léger restant et le placer à l'index 1

Après avoir placé l'œuf le plus léger à l'index 0, l'algorithme cherche l'œuf à placer à l'index 1. A cet effet l'algorithme recherche l'index de l'œuf le plus léger des œufs non triés.

Index actuel (i == 1)	Index du minimum iMin	Œufs triés		Œufs non triés				
Index	0	1	2	3	4	5	6	7
Poids de l'œuf	1	8	13	3	10	15	1	8

A nouveau les œufs à l'index actuel et à l'index du minimum sont échangés.

Index actuel (i == 1)	Index du minimum iMin	Œufs triés		Œufs non triés				
Index	0	1	2	3	4	5	6	7
Poids de l'œuf	1	1	13	3	10	15	8	8

### Troisième étape - Trouver l'œuf le plus léger restant et le placer à l'index 2

La procédure est répétée pour l'index 2. On cherche d'abord l'index de l'œuf le plus léger des œufs non triés.

Index actuel (i == 2)	Index du minimum iMin	Œufs triés		Œufs non triés				
Index	0	1	2	3	4	5	6	7
Poids de l'œuf	1	1	13	3	10	15	8	8

Puis on échange les œufs à l'**index actuel** et à l'**index du minimum**.

<b>Index actuel (i == 2)</b>	<b>Index du minimum iMin</b>		<b>Œufs triés</b>		<b>Œufs non triés</b>			
<b>Index</b>	0	1	2	3	4	5	6	7
<b>Poids de l'œuf</b>	1	1	3	13	10	15	8	8

### Etapas suivantes

L'algorithme procède de la même façon pour les autres index de la liste. Il cherche à chaque étape l'index de l'œuf le plus léger restant, puis échange l'œuf à cet index et l'œuf à l'index actuel.

On peut s'arrêter à l'avant-dernier index. Après avoir placé le bon œuf à l'avant-dernier index, il ne reste plus qu'un seul œuf qui se trouve automatiquement au bon endroit :

<b>Index actuel (i == 6)</b>	<b>Œufs triés</b>		<b>Œuf automatiquement trié</b>					
<b>Index</b>	0	1	2	3	4	5	6	7
<b>Poids de l'œuf</b>	1	1	3	8	8	10	13	15

### Remarque

Pour trier la liste des œufs par ordre décroissant on procède de la même façon, mais on cherche à chaque étape l'index de l'œuf le plus lourd non trié.