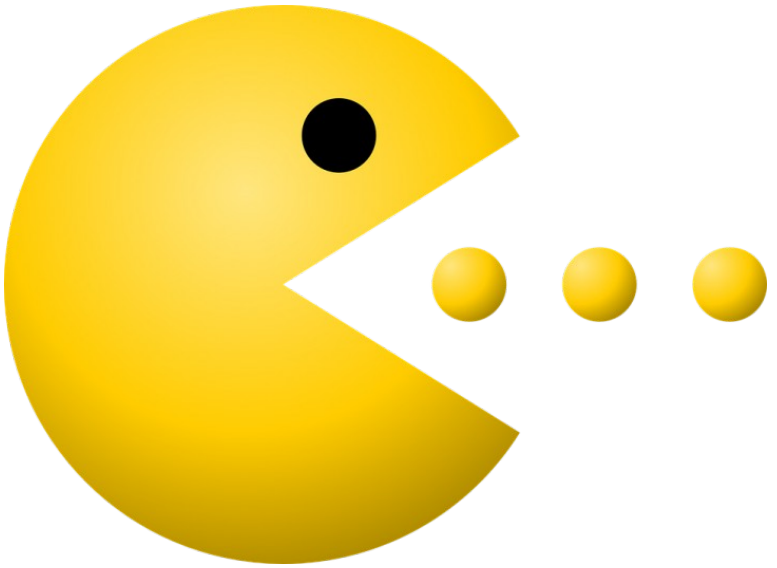
# Les algorithmes gloutons



# Les algorithmes glouton

***Algorithme Glouton (Greedy en anglais : Vorace)***

**Cette méthode offre une solution rapide afin d'optimiser la résolution de problème comportant un très grand nombre de solutions.**

**Cependant la solution trouvée n'est pas forcément optimale**

# Les algorithmes glouton

**Cette méthode construit une solution d’une manière pas à pas. A chaque étape, elle prend la direction la plus prometteuse.**

# Les algorithmes glouton

**Principe 1 : Faire toujours un choix localement optimal dans l'espoir que ce choix mènera à une solution globalement optimale**

# Les algorithmes glouton

**Principe 2 : Ne jamais remettre en cause une décision prise auparavant.**

# Algorithme du rendu de monnaie

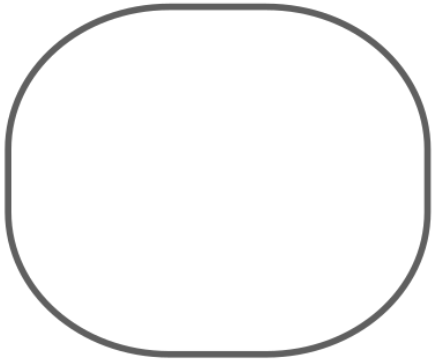
# Les algorithmes glouton

```
piecesDispoDecroissant = [2,1,0.5,0.2,0.1,0.05]
aRendre = 0.45
pour chaque piece de piecesDispoDecroissant :
        nbrePieces = aRendre // piece
        Distribuer nbrePieces
        aRendre = aRendre-nbrePieces x piece
```

# Les algorithmes glouton

```
piecesDispoDecroissant = [2,1,0.5,0.2,0.1,0.05]
aRendre = 0.45
pour chaque piece de piecesDispoDecroissant :
        nbrePieces = aRendre // piece
        Distribuer nbrePieces
        aRendre = aRendre-nbrePieces x piece
```

Piéces rendues



# Les algorithmes glouton

```
piecesDispoDecroissant = [2,1,0.5,0.2,0.1,0.05]
aRendre = 0.45
pour chaque piece de piecesDispoDecroissant :
        nbrePieces = aRendre // piece
        Distribuer nbrePieces
        aRendre = aRendre-nbrePieces x piece
```


Piéces rendues

# Les algorithmes glouton

```
piecesDispoDecroissant = [2,1,0.5,0.2,0.1,0.05]
aRendre = 0.45
pour chaque piece de piecesDispoDecroissant :
        nbrePieces = aRendre // piece
        Distribuer nbrePieces
        aRendre = aRendre-nbrePieces x piece
```


Piéces rendues

# Les algorithmes glouton

```
piecesDispoDecroissant = [2,1,0.5,0.2,0.1,0.05]
aRendre = 0.45
pour chaque piece de piecesDispoDecroissant :
        nbrePieces = aRendre // piece
        Distribuer nbrePieces
        aRendre = aRendre-nbrePieces x piece
```


Piéces rendues

# Les algorithmes glouton

```
piecesDispoDecroissant = [2,1,0.5,0.2,0.1,0.05]
aRendre = 0.45
pour chaque piece de piecesDispoDecroissant :
        nbrePieces = aRendre // piece
        Distribuer nbrePieces
        aRendre = aRendre-nbrePieces x piece
```


Piéces rendues

# Les algorithmes glouton

```
piecesDispoDecroissant = [2,1,0.5,0.2,0.1,0.05]
aRendre = 0.45
pour chaque piece de piecesDispoDecroissant :
        nbrePieces = aRendre // piece
        Distribuer nbrePieces
        aRendre = aRendre-nbrePieces x piece
```


Piéces rendues

# Les algorithmes glouton

```
piecesDispoDecroissant = [2,1,0.5,0.2,0.1,0.05]
aRendre = 0.45
pour chaque piece de piecesDispoDecroissant :
        nbrePieces = aRendre // piece
        Distribuer nbrePieces
        aRendre = aRendre-nbrePieces x piece
```


Piéces rendues

# Les algorithmes glouton

```
piecesDispoDecroissant = [2,1,0.5,0.2,0.1,0.05]
aRendre = 0.45
pour chaque piece de piecesDispoDecroissant :
        nbrePieces = aRendre // piece
        Distribuer nbrePieces
        aRendre = aRendre-nbrePieces x piece
```


Piéces rendues

# Les algorithmes glouton

```
piecesDispoDecroissant = [2,1,0.5,0.2,0.1,0.05]
aRendre = 0.45
pour chaque piece de piecesDispoDecroissant :
        nbrePieces = aRendre // piece
        Distribuer nbrePieces
        aRendre = aRendre-nbrePieces x piece
```


Piéces rendues

# Les algorithmes glouton

```
piecesDispoDecroissant = [2,1,0.5,0.2,0.1,0.05]
aRendre = 0.45
pour chaque piece de piecesDispoDecroissant :
        nbrePieces = aRendre // piece
        Distribuer nbrePieces
        aRendre = aRendre-nbrePieces x piece
```


Piéces rendues

# Les algorithmes glouton

```
piecesDispoDecroissant = [2,1,0.5,0.2,0.1,0.05]
aRendre = 0.45
pour chaque piece de piecesDispoDecroissant :
        nbrePieces = aRendre // piece
        Distribuer nbrePieces
        aRendre = aRendre-nbrePieces x piece
```


Piéces rendues

# Les algorithmes glouton

```
piecesDispoDecroissant = [2,1,0.5,0.2,0.1,0.05]
aRendre = 0.45
pour chaque piece de piecesDispoDecroissant :
        nbrePieces = aRendre // piece
        Distribuer nbrePieces
        aRendre = aRendre-nbrePieces x piece
```


Piéces rendues

# Les algorithmes glouton

```
piecesDispoDecroissant = [2,1,0.5,0.2,0.1,0.05]
aRendre = 0.45
pour chaque piece de piecesDispoDecroissant :
        nbrePieces = aRendre // piece
        Distribuer nbrePieces
        aRendre = aRendre-nbrePieces x piece
```


Piéces rendues

# Les algorithmes glouton

```
piecesDispoDecroissant = [2,1,0.5,0.2,0.1,0.05]
aRendre = 0.45
pour chaque piece de piecesDispoDecroissant :
        nbrePieces = aRendre // piece
        Distribuer nbrePieces
        aRendre = aRendre-nbrePieces x piece
```


Piéces rendues

# Les algorithmes glouton

```
piecesDispoDecroissant = [2,1,0.5,0.2,0.1,0.05]
aRendre = 0.45
pour chaque piece de piecesDispoDecroissant :
        nbrePieces = aRendre // piece
        Distribuer nbrePieces
        aRendre = aRendre-nbrePieces x piece
```


Piéces rendues

# Les algorithmes glouton

```
piecesDispoDecroissant = [2,1,0.5,0.2,0.1,0.05]
aRendre = 0.45
pour chaque piece de piecesDispoDecroissant :
        nbrePieces = aRendre // piece
        Distribuer nbrePieces
        aRendre = aRendre-nbrePieces x piece
```


Piéces rendues

# Les algorithmes glouton

```
piecesDispoDecroissant = [2,1,0.5,0.2,0.1,0.05]
aRendre = 0.45
pour chaque piece de piecesDispoDecroissant :
        nbrePieces = aRendre // piece
        Distribuer nbrePieces
        aRendre = aRendre-nbrePieces x piece
```


Piéces rendues

# Les algorithmes glouton

```
piecesDispoDecroissant = [2,1,0.5,0.2,0.1,0.05]
aRendre = 0.45
pour chaque piece de piecesDispoDecroissant :
        nbrePieces = aRendre // piece
        Distribuer nbrePieces
        aRendre = aRendre-nbrePieces x piece
```


Piéces rendues

# Les algorithmes glouton

```
piecesDispoDecroissant = [2,1,0.5,0.2,0.1,0.05]
aRendre = 0.45
pour chaque piece de piecesDispoDecroissant :
        nbrePieces = aRendre // piece
        Distribuer nbrePieces
        aRendre = aRendre-nbrePieces x piece
```


Piéces rendues

# Les algorithmes glouton

```
piecesDispoDecroissant = [2,1,0.5,0.2,0.1,0.05]
aRendre = 0.45
pour chaque piece de piecesDispoDecroissant :
        nbrePieces = aRendre // piece
        Distribuer nbrePieces
        aRendre = aRendre-nbrePieces x piece
```


Piéces rendues

# Les algorithmes glouton

```
piecesDispoDecroissant = [2,1,0.5,0.2,0.1,0.05]
aRendre = 0.45
pour chaque piece de piecesDispoDecroissant :
        nbrePieces = aRendre // piece
        Distribuer nbrePieces
        aRendre = aRendre-nbrePieces x piece
```


Piéces rendues

# Les algorithmes glouton

```
piecesDispoDecroissant = [2,1,0.5,0.2,0.1,0.05]
aRendre = 0.45
pour chaque piece de piecesDispoDecroissant :
        nbrePieces = aRendre // piece
        Distribuer nbrePieces
        aRendre = aRendre-nbrePieces x piece
```


Piéces rendues

# Les algorithmes glouton

```
piecesDispoDecroissant = [2,1,0.5,0.2,0.1,0.05]
aRendre = 0.45
pour chaque piece de piecesDispoDecroissant :
        nbrePieces = aRendre // piece
        Distribuer nbrePieces
        aRendre = aRendre-nbrePieces x piece
```


Piéces rendues

# Les algorithmes glouton

```
piecesDispoDecroissant = [2,1,0.5,0.2,0.1,0.05]
aRendre = 0.45
pour chaque piece de piecesDispoDecroissant :
        nbrePieces = aRendre // piece
        Distribuer nbrePieces
        aRendre = aRendre-nbrePieces x piece
```


Piéces rendues

# Les algorithmes glouton

```
piecesDispoDecroissant = [2,1,0.5,0.2,0.1,0.05]
aRendre = 0.45
pour chaque piece de piecesDispoDecroissant :
        nbrePieces = aRendre // piece
        Distribuer nbrePieces
        aRendre = aRendre-nbrePieces x piece
```


Piéces rendues

# Les algorithmes glouton

```
piecesDispoDecroissant = [2,1,0.5,0.2,0.1,0.05]
aRendre = 0.45
pour chaque piece de piecesDispoDecroissant :
        nbrePieces = aRendre // piece
        Distribuer nbrePieces
        aRendre = aRendre-nbrePieces x piece
```

*piecesDispoDecroissant*

| 2 | 1 | 0.5 | 0.2 | 0.1 | 0.05 |
|---|---|---|---|---|---|

| *aRendre* | *piece* | *nbrePieces* |
|---|---|---|
| 0.05 | 0.05 | 1 |

Piéces rendues

# Les algorithmes glouton

```
piecesDispoDecroissant = [2,1,0.5,0.2,0.1,0.05]
aRendre = 0.45
pour chaque piece de piecesDispoDecroissant :
        nbrePieces = aRendre // piece
        Distribuer nbrePieces
        aRendre = aRendre-nbrePieces x piece
```

*piecesDispoDecroissant*

| 2 | 1 | 0.5 | 0.2 | 0.1 | 0.05 |
|---|---|---|---|---|---|

| *aRendre* | *piece* | *nbrePieces* |
|---|---|---|
| 0.0 | 0.05 | 1 |

Piéces rendues

# Les algorithmes glouton

```
piecesDispoDecroissant = [2,1,0.5,0.2,0.1,0.05]
aRendre = 0.45
pour chaque piece de piecesDispoDecroissant :
        nbrePieces = aRendre // piece
        Distribuer nbrePieces
        aRendre = aRendre-nbrePieces x piece
```

*piecesDispoDecroissant*

| 2 | 1 | 0.5 | 0.2 | 0.1 | 0.05 |
|---|---|---|---|---|---|

| *aRendre* | *piece* | *nbrePieces* |
|---|---|---|
| 0.0 | 0.05 | 1 |

Piéces rendues

# Synthèse

# Les algorithmes gloutons

**Dans le contexte de la machine à rendre la monnaie, il y aura :**

- Au plus une pièce de 5 cents sinon une pièce de 10 cents.
- Au plus un pièce de 10 cents sinon une de 20 cents
- Au plus deux pièces de 20 cents sinon une de 50 cents
- Au plus une pièce de 50 cents sinon une de 1€
- Au plus une pièce de 1€ sinon une de 2€

**Donc l'algorithme est optimal**

# Les algorithmes gloutons

**Pour mettre au point un algorithme glouton, il faut donc:**

- **Trouver un critère de sélection.**
  **Souvent facile ... mais pas toujours**
- **Montrer que le critère est bon, c'est à dire que la solution obtenue est une des meilleures voir la meilleure**
  **Souvent difficile !**
- **L'implémenter.** **En général facile et efficace**