

Conception d'algorithmes

Retour Arrière

Le problème du sac à dos

- Vu l'année dernière (cours + tutorat)
- .. ?
- Rappel sur les algo récursifs ...
 - ??

Ecriture d' un algorithme récursif

➤ Ingrédients:

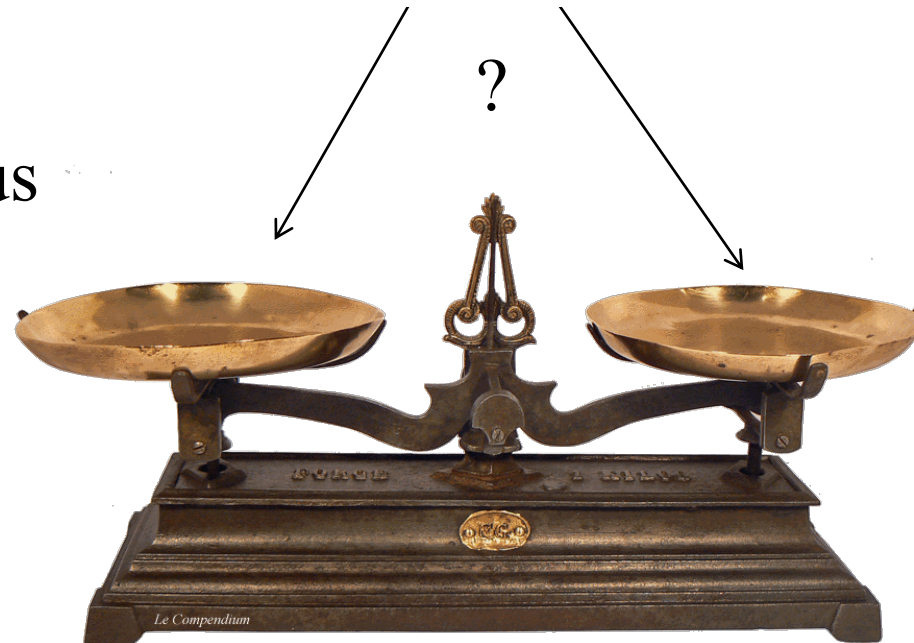
- Cas trivial
- Cas récursif
- Quantité de contrôle (assurer terminaison)
- Paramétrage du problème et premier appel

Exemple1: équilibrer une balance à plateau

Un ensemble de n poids
 w_1, w_2, \dots, w_n



Comment les
répartir en 2 sous
ensemble de
masses égales



Formulation

- Soit $x_i = 0$ si la masse i est à gauche, et $x_i=1$ sinon
- Objectif:

$$\sum_{i=1}^n x_i w_i = \sum_{i=1}^n (1 - x_i) w_i$$

- Ou minimiser

$$\delta = \sum_{i=1}^n x_i w_i - \sum_{i=1}^n (1 - x_i) w_i$$

Principe de l'algorithme

➤ Principe récursif

- On prend le « premier » poids, on l'affecte à un plateau et on regarde si ça peut donner la solution en passant au suivant
- Sinon on affecte ce « premier » à l'autre plateau et on regarde si ça peut donner la solution en passant au suivant

➤ Cas trivial:

- On a affecté tous les poids ➔ on retourne vrai si la différence des plateau est nulle

➤ Paramétrage

- Le rang du poids considéré

Implantation

- Structures de données nécessaires
- Algorithme
- Comment afficher la solution (où est elle?)
- Passage à un problème d'optimisation ?

Algorithme

- ???
- À vous

Age Group	Percentage
18-24	15%
25-34	20%
35-44	25%
45-54	20%
55-64	15%
65-74	10%
75-84	5%
85+	5%

- V. Chevrer

Algorithme moins informel

- resoudre (n: nœud): booléen
 si feuille(n) alors retourner solution(n)
 sinon
 pour c dans fils(n) faire
 Si resoudre(c) alors retourner VRAI fsi
 fpour
 retourner FAUX
 fsi
- solution(n: nœud): booléen
 Indique si n est solution

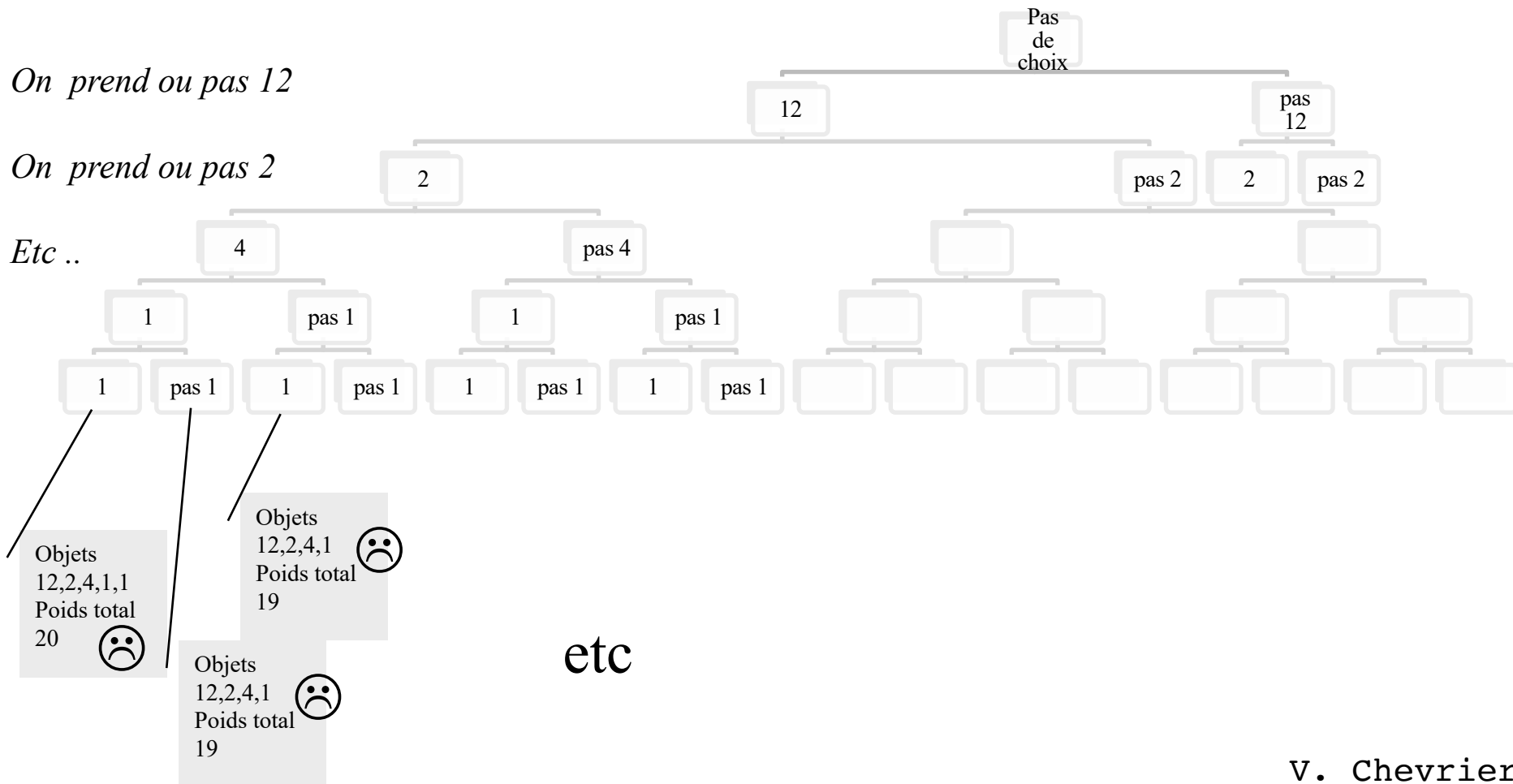
Illustration avec le problème du sac à dos

charge = 15
paquets = 12, 2, 4, 1, 1

On prend ou pas 12

On prend ou pas 2

Etc ..



Remarques

- Arrêt
 - Quand solution trouvée (résolution de problème)
 - Quand optimum trouvé (optimisation) donc lorsque tout est parcouru
- Ici description d'un parcours en profondeur d'abord
 - Possibilité d'ordonner les fils selon une heuristique
- Parfois on trouve des états déjà rencontrés
 - Comment éviter de refaire les calculs ...
 - Équivalent à un parcours de graphe
 - Optimisation = plus court chemin ...

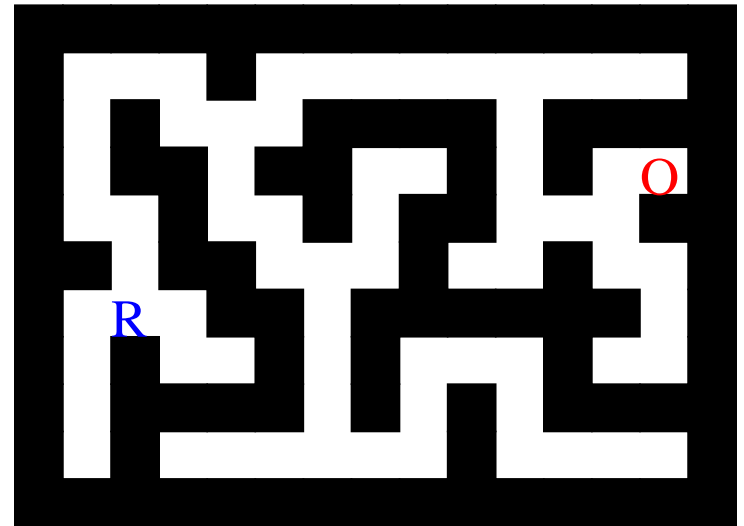
Cas d'application

➤ Exemples

- Cavalier sur échiquier
- Balance à N plateaux
- Trouver un chemin/un objet dans un labyrinthe

➤ Cadrage

- Représentation du problème
- Principe d'algorithme récursif



```
private boolean resoudreR(int i) {  
    // trivial cases  
    if (i == nbMass) {  
        return sommeOk();  
    } else {  
        // here we try all the possibilities  
        for (int possibility = 0; possibility < nbSet; possibility++) {  
            repartition[i - 1] = possibility;  
            if (resoudreR(i + 1)) // on essaie en mettant sur le ieme  
                                // plateau)  
            {  
                return true;  
            }  
        }  
        // all is tried and fails  
        return false;  
    }  
}
```