

Cours d'algorithmique

IUT Informatique première année
Olivier ROUSSEL
olivier.rousseau@iut-lens.univ-artois.fr

Introduction générale

Les modalités du cours

La matière

- Matière fondamentale en informatique
- Nécessité de la maîtriser pour pouvoir aborder sereinement les autres matières
- Gros volume horaire (1,5 H cours, 3H TD, 1,5H TP par semaine)
- N'hésitez pas à poser des questions : en cours, en TD et en TP ou par mail !

Votre nouveau statut

- Vous êtes maintenant des étudiants !
- C'est à vous de vous prendre en main !
- C'est à vous d'organiser votre prise de notes (surtout pas de recopie complète !)
- C'est à vous d'organiser votre travail
- C'est à vous de gérer vos révisions

Votre nouveau statut (suite)

- C'est à vous de compléter les cours par des exercices, TP, lectures personnelles (CDI,...)
- C'est à vous de poser des questions aux enseignants
- C'est à vous de construire votre avenir !

Soyez actifs !

- Vous aurez souvent des copies des transparents du cours. Certaines précisions ne seront données que par oral. Il faut donc compléter le support de cours par vos annotations personnelles.
- Vous devez **apprendre votre cours régulièrement !**
- N'attendez pas en TD que la réponse soit donnée
- Cherchez *votre* solution ! Comparez la avec celle de votre voisin (**sauf en DS !!!**) et la correction
- Posez des questions chaque fois que vous avez un doute
- Ne laissez rien passer

Le contrôle des connaissances

- Contrôle continu : 2/3 de la note
 - 3 devoirs surveillés (voir transparent suivant)
 - environ 3 contrôles de travaux pratiques ou projets
 - **des contrôles surprises réguliers en cours, TD ou TP.**
- L'ensemble des TP et contrôles surprises aura à peu près le même poids qu'un DS (selon difficulté)
- Un partiel : 1/3 de la note
- Ne vous contentez pas de la moyenne ! Personne n'est à l'abri d'une défaillance, ni ne connaît la difficulté des épreuves à venir.

Document autorisé en DS

- Une seule page au format A4, recto/verso
- Uniquement manuscrite (pas de photocopie, ni d'impression !)
- Doit impérativement porter votre nom.

Le soutien

- Des séances de soutien seront organisées.
- Elles auront lieu dans l'un des créneaux libre (samedi matin par exemple)
- Elles seront obligatoires pour ceux qui auront obtenu les x plus mauvaises notes au contrôle précédent.

Les projets et TP

- Les TP et les projets nécessitent une préparation personnelle
- Il faut avoir réfléchi au problème qui vous est posé AVANT de venir en séance de TP travailler sur la machine. Vous devez donc avoir préparé une solution en pseudo-langage **avant la séance**.
- Sachez planifier votre activité : vous aurez forcément plusieurs TP en même temps ou à rendre quasiment en même temps

La remise des TP

- On vous demandera souvent de rendre un fichier
- Comme pour une copie, le fichier doit être rendu au professeur et non pas être conservé dans votre serviette
- Prenez donc grand soin de rendre votre fichier à l'endroit que l'on vous indique et dans tous les cas dans le répertoire ~/prof
- Sinon vous aurez ZERO !!

Introduction à l'algorithmique et à la programmation

Une comparaison

- On pourra souvent comparer le travail que l'on va effectuer en algorithmique avec une recette de cuisine.
- Une recette de cuisine permet de **transformer des ingrédients** en un **produit fini** par une **succession de manipulations** qui doivent s'effectuer dans un **ordre précis**.

Une recette

Phase	Op	Titre	Déroulement	Denrées	Ustensiles
100	10	Préparation du matériel	Installer et vérifier le matériel	½ l de lait, 50g de farine, 75g de sucre, 1 œuf entier,	1 casserole, 1 saladier, 1 spatule, 1 verre doseur, 2 bols
	20	Pesée des denrées	Préparer et peser les denrées		
200	10	Ébullition du lait	Porter à ébullition le lait avec l'arôme choisi	Lait, arôme	Casserole
300	10	Sucre et œuf	Verser le sucre sur les œufs. Travailler pour que le mélange blanchisse	Œufs, sucre	Saladier, spatule
	20	Incorporation farine			

L'ordre à respecter

Le récipient

La manipulation

L'ingrédient que l'on transforme

Une recette de cuisine

Phase	Op	Titre	Déroulement	Denrées	Ustensiles
			mélanger vigoureusement	farine, lait	
500	10	Cuisson	Mettre le mélange dans une casserole. Faire cuire sans cesser de tourner. Atteindre l'ébullition. Faire épaissir au feu. Ôter du feu.	Crème	Casserole, spatule
600	10	Rangement	Ranger les denrées et grouper les ustensiles		
	20	Vaisselle	Effectuer la vaisselle		Nécessaire vaisselle
	30	Nettoyage	Ranger le matériel et nettoyer le poste de travail		

Une opération à répéter pour atteindre un résultat

Le regroupement d'opérations

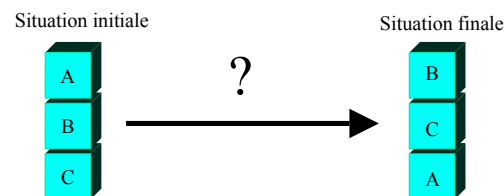
Le nettoyage après le travail

Algorithme

- Définition du dictionnaire : calcul, enchaînement des actions nécessaires à l'accomplissement d'une tâche
- Plus simplement : méthode permettant de passer d'une situation initiale à une situation finale

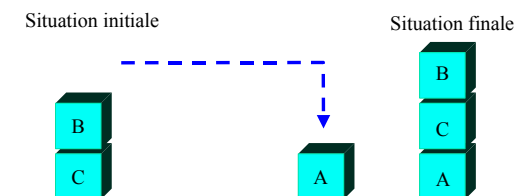
Exemple simpliste d'algorithme

- Des piles de cubes :



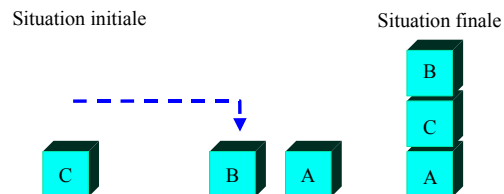
Étape 1

- Des piles de cubes :



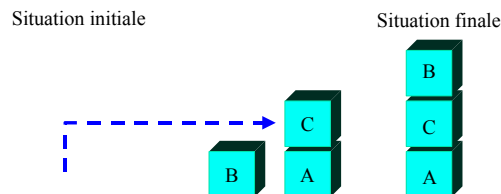
Étape 2

- Des piles de cubes :



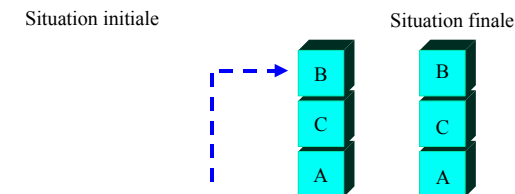
Étape 3

- Des piles de cubes :



Étape 4

- Des piles de cubes :



Pouvoir communiquer la solution

- On peut utiliser un langage bavard :
 - Déplacer le bloc A du tas initial sur le tas final
 - Déplacer le bloc B du tas initial sur un tas intermédiaire
 - Déplacer le bloc C du tas initial sur le tas final
 - Déplacer le bloc B du tas intermédiaire sur le tas final

Pouvoir communiquer la solution

- ou utiliser un langage plus concis, plus mathématique et en anglais :
 - Move(0,2);
 - Move(0,1);
 - Move(0,2);
 - Move(1,2);

L'addition,...

- Vous connaissez des algorithmes depuis bien longtemps !

$$\begin{array}{r} \overset{1}{5} \overset{1}{6} \\ + 65 \\ \hline = 121 \end{array}$$

...et les autres opérations

- Savez vous soustraire, multiplier et diviser à la main ?
- Sauriez vous expliquer la méthode (l'algorithme) à utiliser à
 - l'un de vos camarades ?
 - un enfant de 6 ans ?
 - l'ordinateur ?

Recette de cuisine

- Un algorithme peut être comparé à une recette de cuisine
- Une recette est la méthode à suivre pour transformer les denrées en un produit fini
- Un algorithme est une méthode à suivre pour transformer une situation initiale en une situation finale

Programme

- Un programme est une **séquence** d'opérations à effectuer pour passer d'une information initiale à un résultat final

Phases d'un programme (1)

- Un programme comportera 3 phases :
- **L'initialisation** : ensemble des opérations à effectuer au tout début pour placer la machine dans un état parfaitement connu à partir duquel on peut commencer le traitement de l'information
(=préparer les ustensiles, nettoyer le plan de travail)

Phases d'un programme (2)

- **Le traitement** : ensemble des opérations qui permettent de passer de l'information initiale au résultat attendu
(=la recette de cuisine)

Phases d'un programme (3)

- **La finalisation** : ensemble des opérations à effectuer à la fin d'un programme pour revenir à un état parfaitement connu qui ne gênera pas l'exécution des programmes qui suivent
(=la vaisselle)

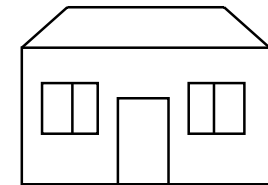
Comparaison avec l'architecture

- Programmer, c'est un peu comme utiliser des briques de bases à notre disposition (parpaing ? briques ?) pour construire une usine qui transformera les données comme on le souhaite

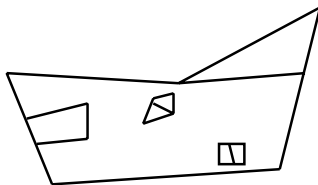
Art ou technique ?

- La programmation, comme la cuisine, comme l'architecture est un art. Donc, pas de recette miracle en général.
- La programmation, comme l'architecture est aussi une technique qui ne tolère pas la moindre erreur (la cuisine non plus)
- Il est nécessaire de laisser un plan pour permettre d'étendre le programme

Une maison



Une maison ?



Différence entre ces maisons

- Dans les deux cas, il y a une porte, deux fenêtres et un toit, ce qui répond bien à une définition de « maison »
- Mais... l'une est solide, confortable et facilement extensible tandis que l'autre est fragile et difficilement utilisable ou aménageable
- Payeriez-vous le même prix pour les deux ?

Notation des programmes

- Vos programmes seront notés selon plusieurs critères :
 - Résolvent-ils le problème qui vous a été posé ?
 - Le résolvent-ils de manière élégante et facilement adaptable pour résoudre un problème proche ?
 - Avez-vous laissé un plan ?

Les commentaires

- La première manière de laisser un « plan » pour ceux qui reliront et utiliseront vos programmes (VOUS le premier !!) est de commenter votre programme, c'est à dire d'indiquer en français ce que vous avez voulu faire, ce que fait telle ou telle partie de votre programme, à quoi sert telle ou telle variable

Signez vos chefs-d'œuvre

- Votre programme doit commencer par le commentaire suivant (en-tête) (-1 pt si absent):
/*
 nom: Jean
 prénom: DUPONT
 groupe: 1.2
 titre: résolution d'équations
 date: 20/10/2001
*/

Nécessité d'une méthodologie

- L'écriture d'un programme (tout comme la construction d'un bâtiment) nécessite de suivre une méthodologie

Méthodologie (1)

- **Analyse des besoins :**
quel est exactement le problème que l'on veut résoudre ?
quel est exactement le bâtiment à construire (Cabane, immeuble, château ? Combien de pièces ?)

Méthodologie (2)

- **Spécifications fonctionnelles :**
quelle relation doit-il y avoir entre l'entrée et la sortie du programme ?
(N.B. on considère le programme comme une boîte noire et on s'intéresse à la fonction qui relie la sortie à l'entrée de la boîte)
quels sont les services que le bâtiment doit rendre ?

Méthodologie (3)

- **Spécifications non fonctionnelles :**
quelles contraintes pour réaliser la boîte noire qui effectue le traitement ?
Quels matériaux utiliser ? Combien de temps pour construire ? Quel budget ?

Méthodologie (4)

- **Analyse :**
décomposition plus ou moins détaillée du problème en sous problèmes (on utilise un langage algorithmique ou pseudo langage)
plan du bâtiment à une échelle plus ou moins détaillée

Méthodologie (5)

- **Codage :**
transcription en langage informatique
construction du bâtiment

Méthodologie (6)

- **Tests :**
vérification du fonctionnement du programme
contrôle du travail

Méthodologie (7)

- **Documentation :**
mode d'emploi, documents de maintenance
mode d'emploi, plans, guide d'entretien

Adapter la méthode

- Bien sûr, on ne construit pas une cabane dans un arbre en prenant autant de précautions que pour construire un immeuble
- Mais dans tous les cas, il faut que le résultat satisfasse le client

Comment analyser un problème ?

- Utiliser l'analyse descendante :
décomposer un problème en sous problèmes plus simples à résoudre et recommencer jusqu'à ce qu'on arrive à des problèmes que l'on sache résoudre avec les briques de base à notre disposition

Programmer

- Programmer, c'est **analyser un problème** pour trouver la séquence d'opérations qui permet de le résoudre puis transcrire ces opérations dans un langage informatique.
- Il faut d'abord savoir résoudre un problème « à la main » avant de chercher à le programmer ! (on ne peut pas apprendre à quelqu'un ce qu'on ne sait pas soi-même)

Les langages informatiques

- Une multitude de langages, chacun étant bien adapté à la résolution de certains problèmes
- Seulement 3+1 grandes familles de langages
- Seulement une poignée de langages génériques et largement répandus

Différents exemples de langages

- Les programmes qui suivent effectuent tous la même opération : afficher le mot « bonjour » à l'écran et passer à la ligne
- Exemples en
 - Langage machine, assembleur
 - C, C++, Java
 - Pascal, Ada
 - PostScript
 - Lisp
 - Prolog

Langage machine

```
6f75 720a 0000 4743
2065 6763 732d 322e
3939 3033 3134 2f4c
6373 2d31 2e31 2e32
2900 002e 7379 6d74
6162 002e 7368 7374
7874 002e 7265 6c2e
7461 002e 6273 7300
6f64 6174 6100 2e63
```

Assembleur

```
.section      .rodata
.LC0:
.string "Bonjour\n"

.text
.align 4
.globl main
.type        main,@func

main:
    pushl %ebp
    movl %esp,%ebp
    pushl $.LC0
    call printf
    addl $4,%esp
    leave
    ret
```

Le langage C

```
#include <stdio.h>
int main()
{
    printf("Bonjour\n");
}
```

Le langage C++

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Bonjour\n";
}
```

Java

```
class Essai
{
    public static void
        main(String args[])
    {
        System.out.print
            ("Bonjour\n");
    }
}
```

Pascal

```
program Essai;
begin
    writeln('Bonjour');
end.
```

Ada

```
with ada.text_io;
use ada.text_io;

procedure Essai is
begin
    put("Bonjour");
    new_line;
end Essai;
```

PostScript

```
%!
/Helvetica findfont 12
    scalefont setfont
50 50 moveto
(Bonjour) show
showpage
```

Lisp

```
(write "bonjour")
```

Prolog

```
all :- write("Bonjour"),nl.
```

Les grandes familles de langages

- **Langage procédural** : un programme est décrit par une succession d'actions à effectuer
- **Langage fonctionnel** : un programme est décrit par une composition de fonctions à appliquer
- **Langage logique** : un programme est décrit par des buts à atteindre et les sous-buts qui doivent être atteints pour aboutir à ces buts

Langages objets

- Dans un langage objet, les informations et les opérations qui permettent de les manipuler sont regroupées ensemble dans ce qu'on appelle un objet
- Un objet est une sorte de petite usine indépendante qui effectue une partie des transformations du programme.

Quel langage ?

- On se restreindra aux langages procéduraux
- Parmi les plus connus : Basic, Pascal, Ada, C, C++, Java...
- Ces langages possèdent tous des structures communes

Esperanto ?

- Face à tant de langages, on souhaiterait avoir un langage commun pour présenter un algorithme et être compris par tous sans s'enfermer dans un langage donné
- Diverses solutions :
 - Organigramme
 - Arbre programmatique
 - **Pseudo-langage**

Le pseudo-langage

- Le pseudo langage est un langage simple qui se veut assez proche du français tout en introduisant la rigueur nécessaire à l'analyse informatique
- C'est grossièrement un Pascal francisé dans lequel on peut s'autoriser certaines libertés
- Inutilisable par la machine mais très pratique pour expliquer un algorithme

Quel langage sur machine ?

- On utilisera le langage C++ mais sans faire intervenir les notions d'objet dans un premier temps