

# Analyse de ressources en algorithmique et programmation au collège et au lycée

Proposition de critères d'analyse  
et applications sur des exemples de ressources

*10 mars 2017*

*Présentation à une demi-journée de travail commune  
Aux CII Didactique et CII Lycée  
Paris*

Simon Modeste, IMAG, Université de Montpellier  
[simon.modeste@umontpellier.fr](mailto:simon.modeste@umontpellier.fr)

# Introduction

- Quelques critères pour analyser les ressources en algorithmique et programmation
- Collège / Lycée, Mathématiques / Informatique
- Objectifs et connaissances en jeu dans les situations proposées ?
- Objectif : faire apparaître les choix (point de vue de l'institution, point de vue de l'apprentissage)
- Détour épistémologique (savoir de référence)
- Prise en compte du discours et des activités
- Organisation des savoirs et façon dont on les aborde

# Plan

- Algorithmique et programmation
- Organisations des savoirs, organisations didactiques
- Une « grille » pour analyser les ressources
- Essai d'application à différentes ressources
  - Lycée : programmes et documents ressources
  - Cycle 4 : programmes et documents ressources
  - Cycle 4 : manuels (transmaths, sésamaths)
  - Un détour par l'Ukraine
  - Des activités débranchées

# Algorithmique et programmation

# Algorithme

- Définition :

« Un algorithme est une procédure de résolution de problème, s'appliquant à une famille d'instances du problème et produisant, en un nombre fini d'étapes constructives, non-ambiguës et organisées, la réponse au problème pour toute instance de cette famille. »

# Propriétés/caractéristiques

Des caractéristiques du concept algorithme que l'on peut classer selon 5 grands aspects :

- **Résolution de problème**
- **Effectivité**
- Preuve
- Complexité
- Modèles théoriques

# Résolution de problème

Un algorithme résout un problème, c'est-à-dire, répond à une question précise posée pour une famille d'instances. Pour chaque instance, le même algorithme donne une réponse à la question du problème. On parle d'entrée et de sortie de l'algorithme : l'entrée représente l'instance à traiter, la sortie la réponse à la question pour cette instance.

Les propriétés importantes sont :

- les notions d'entrée et de sortie
- La spécifications
- la notion d'instance d'un problème auquel répond l'algorithme
- le fait qu'un algorithme réponde à une question pour toutes les instances du problème

# Effectivité

- Un algorithme apporte une solution effective. C'est une méthode systématique qu'un opérateur peut mettre en œuvre. Sur des données finies, en suivant des règles non-ambiguës, un algorithme s'exécute en un nombre fini d'étapes. L'opérateur peut être de toute sorte mais bien souvent il s'agit d'un ordinateur pour lequel l'algorithme a été exprimé sous forme d'un programme dans un langage extrêmement précis. La notion d'algorithme englobe d'autres notions pouvant être constructives et finies : formules mathématiques, programmes de calcul, constructions géométriques, activité de la vie courante...

Les propriétés importantes sont :

- un algorithme peut être mis en œuvre par un opérateur quelconque,
- un algorithme est exécutable par un ordinateur/une machine,
- un algorithme peut être exprimé sous forme d'un programme,
- un algorithme agit sur des données finies,
- un algorithme s'exécute en un nombre fini d'étapes,
- la non-ambiguïté de la description des étapes,
- un algorithme permet de décrire des formules, des programmes de calcul ou de construction.

# Représentation des algorithmes

- Diversité des représentations des algorithmes :
  - Langage courant, mathématique
  - Langages de programmation
  - Pseudo-code
  - Organigrammes
  - ...

# Programmation

- **Définition :**

Un programme est une suite d'instructions décrite dans un langage de programmation.

Un algorithme peut être décrit sous forme d'un programme dans un langage donné.

Un programme ne décrit pas nécessairement un algorithme.

# Lien algorithmique/programmation

- Le développement de l'informatique et de la programmation a influencé la définition actuelle d'algorithme (Chabert) :
  - Finitude (données, instructions, exécutions)
  - Structures de contrôle
  - Spécifications
  - Variables
  - Complexité
- Paradigmes de programmation
  - Impérative, événementielle, parallèle, orientée objet...
  - Lié à la notion d' « opérateur »
- Opérateurs « machines » et langages spécifiques → programmation

# Quelques remarques sur les structures de contrôle

- On distingue les structures de contrôle suivantes :
  - Instruction conditionnelle : if ... then ... (else ...)
  - Répétition finie : Répéter ... fois ... ; Pour ... allant de ... à ... faire ... ; Pour ... dans ... faire ...
  - Répétition conditionnelle : Tant que ... faire ... ; Faire ... jusqu'à ...
  - (d'autres possibilités devenues désuètes...)
  - (d'autres type d'instructions « équivalentes » dans d'autres paradigmes)
- Raisons d'être de ces structures :
  - « économie »
  - puissance de calcul
  - Bonnes pratiques de programmation (différents niveaux de contraintes) : « pas de pratiques expertes »

Organisations des savoirs, organisations didactiques

# La machine dans l'enseignement de la programmation et de l'algorithmique

- Nguyen montre qu'il existe deux grande approches dans l'enseignement de la programmation/algorithmique :
  - Présence d'une machine de référence (concrète ou abstraite)
  - Présence d'un langage spécifique

Ingénierie didactique pour introduire les boucles et les variables à partir de machines-calculatrice de plus en plus évoluées. Enjeu d' « économie » pour introduire la répétition.

# Psychologie de la programmation

- Prise en compte des aspects cognitifs
- S'intéresse à l'apprentissage de la programmation et l'association (de suites) d'opérations (avec structures de contrôles et variables) pour résoudre un problème ou une situation donnée.
- Diversité d'usages pour une même structure de contrôle

# Situations et activités proposées

- Quels connaissances à construire ? Quels objectifs ?
- Il est donc nécessaire de se pencher sur les tâches proposées aux élèves, leur organisation, les contrôles possibles, les objectifs d'apprentissage fixés et la pertinence de ces tâches pour ces objectifs.
- Quelle institutionnalisation ? Pour quels contenus ?

Une « grille » pour analyser les ressources

# Grille (1)

- Discours général, cours, institutionnalisations, et organisation des savoirs :
  - Quel « concept » d'algorithme ? Quelle définition ? Quelles propriétés et caractéristiques mises en avant dans le discours ? Quel rôle/fonction assigné à l'algorithme dans le discours ?
  - Quels lien entre algorithmique et programmation ?
  - Discours ? Organisation des savoirs ? Machine/Langage/Opérateur de référence ?
  - Quelle progression/hiérarchie/organisation proposée dans ces concepts ?

# Grille (2)

- Exemples et contenus des activités proposées à l'élève (praxis et organisations didactiques) :
  - Comment est organisée la progression dans les concepts enseignés ?
  - Quels algorithmes et quels problèmes (mathématiques ou informatiques) rencontre-t-on ?
  - Quelles tâches proposées à propos de ces algorithmes ? Relèvent-elles de l'algorithmique ou de la programmation ?
  - Quels aspects et caractéristiques (des algorithmes et des programmes) sont mis en jeu dans les tâches proposées ?

# Grille (3)

- À l'échelle d'une activité ou d'une situation :
  - En quoi les problèmes proposés (ou les situations) permettent la construction des concepts visés ? (Nécessité de la connaissance visée pour réussir la tâche)
  - Quel jeu de rétroactions ? Quels choix de variables didactiques ?
  - Spécificités des notions en jeu dans les problème/champs particuliers traités dans l'activité ?
  - Comment les notions abordées sont institutionnalisées (identifiées, décontextualisées, réinvesties...) ?

Essai d'application à différentes ressources

# Non présenté ici

- Programmes et ressources du lycée
- Programmes et documents ressources (cycle 4)
- Manuels du collège (Sesamaths, Transmaths)
- Détour par l'Ukraine... (d'autres choix)
- Activités débranchées (questions didactiques)

# Conclusion et perspectives

- Diversités des choix possibles pour l'enseignement de l'algorithmique et la programmation
- Comment s'appuyer sur l'épistémologie et identifier les savoirs dans les questions didactiques en algorithmique
- À prendre en compte :
  - difficultés des élèves (obstacles)
  - Cohérence de l'organisation des savoirs (dimensions épistémologique)
  - pertinence et effets des choix didactiques (aspects cognitifs)