

Modéliser les ordinateurs et leurs ressources

Manon Blanc

05/01/2026

Visite au collège Jean Rostand

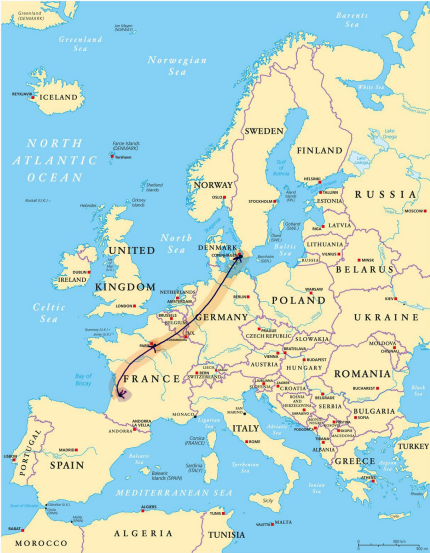
- Mon parcours
- La science des ordinateurs, inventer des machines à calculer
- C'est quoi un algorithme ?
- Calculabilité et Complexité

Qui suis-je ? Mon parcours

BAC	Bac Scientifique (spécialité Mathématiques)
L1 L2	Classes Préparatoires MPSI/MP
L3 M1 M2	École Normale Supérieure de Cachan/ Paris-Saclay (Spécialisation en informatique fondamentale)
Doctorat	Thèse de doctorat en informatique

- Postdoctorat à l'Université de Copenhague

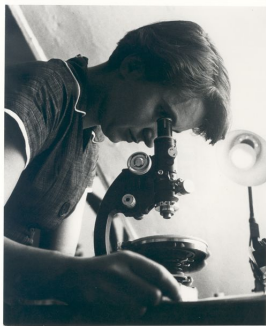
Sur une carte



Source :Peter Hermes Furian – stock.adobe.com

Connaissez-vous les métiers de la recherche ?

Connaissez-vous des scientifiques célèbres ?

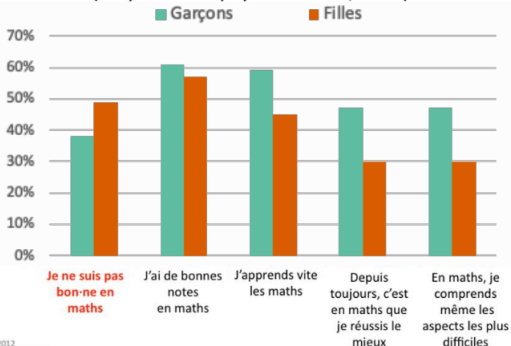


Maria Skłodowska-Curie	Rosalind Franlin	Katherine Coleman-Johnson
------------------------	------------------	---------------------------

Maryam Mirzakhani	Cecilia Payne-Gaposchkin	Émilie du Châtelet
-------------------	--------------------------	--------------------

Quelques chiffres

Attitudes des élèves vis-à-vis des mathématiques (moyenne des pays de l'OCDE, 2012)

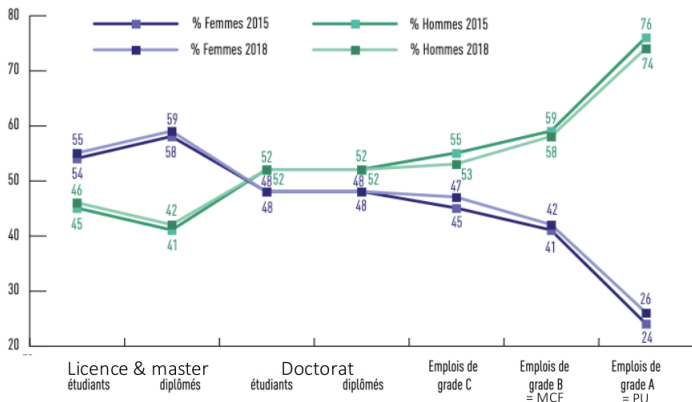


Year: 2012
Source: OECD, 2015

Source : Clémence Perronnet & Manon Réguer-Petit

Quelques chiffres (dans le supérieur)

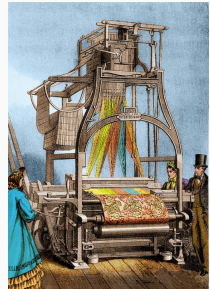
Part des femmes et des hommes au cours d'une carrière universitaire européenne type entre 2015 et 2018



Données pour 28 pays européens. Eurostat, traitement MESRI-SIES.
Rapport *Vers l'égalité femmes-hommes ? Chiffres-clés 2022*, MESRI, p. 14

La science des ordinateurs, inventer des machines à calculer

Ada Lovelace : Une mère fondatrice des ordinateurs (entre autres...)





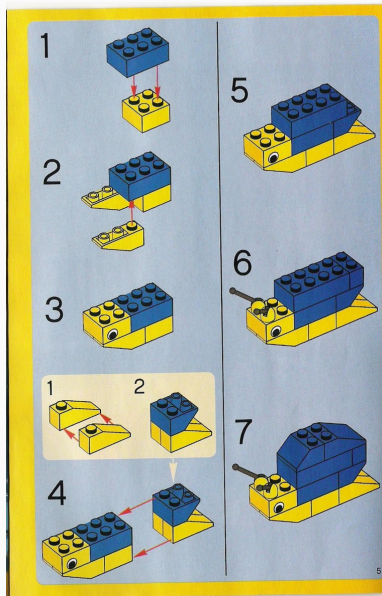
“Mon travail mathématique
implique une imagination
considérable”

**Motivations : c'est quoi un
algorithme ?**

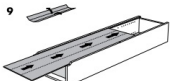
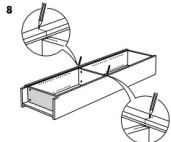
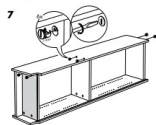
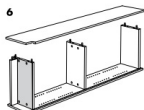
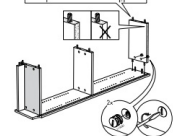
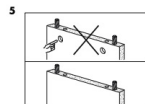
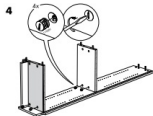
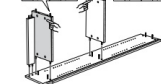
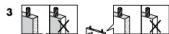
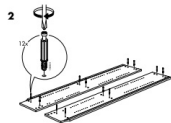
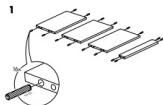
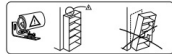
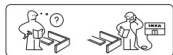
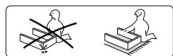
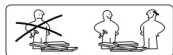
Motivations : c'est quoi un algorithme ?

```
22
23
24 let rec evalexpr context (expr : expr loc) = match expr with
25 |_, Var(v) -> List.assoc v context
26 |_, App(⟦, Lam(var, _, e1)), e2) -> let e2 = evalexpr context e2 in
27   evalexpr ((var, e2)::context) e1
28 |c, App(e1, e2) -> let e1 = evalexpr context e1 in
29   evalexpr context (c, (App(c, e1), e2)))
30 |c, Lam(va, tyop, e1) -> Lam(va, tyop, e1)
31 |c, Pair(e1, e2) -> let e1 = evalexpr context e1 in
32   let e2 = evalexpr context e2 in
33   Pair(c, e1), (c, e2))
34 |c, LetIn(va, e1, e2) -> let e1 = evalexpr context e1 in
35   evalexpr ((va, e1)::context) e2
36 |l1, Fix(l2, (Lam(va, tyop, e1))) -> evalexpr ((va, Fix(l2, Lam (va, tyop, e1)))::context) e1
37 |_, Fix(e1) -> evalexpr context e1
38 |_, Int(n) -> Int(n)
39 |_, Bool(b) -> Bool(b)
40 |_, Proj(ele) -> (match ele with
41 |Left(⟦, Pair(e1, e2)) -> evalexpr context e1
42 |Right(⟦, Pair(e1, e2)) -> evalexpr context e2
43 |_ -> failwith "unprojectable")
44 |_, Ite(e1, e2, e3) -> if (evalexpr context e1) = Bool(true) then
45   evalexpr context e2
46   else
47     evalexpr context e3
48 |l, Binop(bin, e1, e2) -> evalbinop bin context (l, (evalexpr context e1)) (l, (evalexpr context e2))
49 |_, Unit -> Unit
50
51
52
53 let rec eval_cmds context (cmds : cmd loc) = match cmds with
54 |(l, Let(va, tyop, exprloc)) -> (va, (evalexpr context exprloc))
55 |(l, LetRec(va, tyop, exprloc)) -> (va, (evalexpr context (l, (Fix (l, (Lam(va, tyop, exprloc)))))))
56
57
58
59
60 let eval_ast (cmds : Ast.t) = let rec aux context cmds = match cmds with
61 |[] -> context
62 |t::q -> (aux ((eval_cmds context t)::context) q)
63 in
64 aux [] cmds
65
```

Essentiellement, c'est ça



BILLY



Un premier exemple

Entrée : Un entier i

Si : $i \geq 11$

Retourner : “ i est plus grand que 11”

Sinon :

Retourner : “ i n'est pas plus grand que 11”

Et les problèmes commencent à apparaître...

Entrée : Un entier i plus grand que 0

Tant que : $i \geq 0$

$i \leftarrow i + 1$

Retourner : La valeur de i

Et les problèmes commencent à apparaître...

Entrée : Un entier i plus grand que 0

Tant que : $i \geq 0$

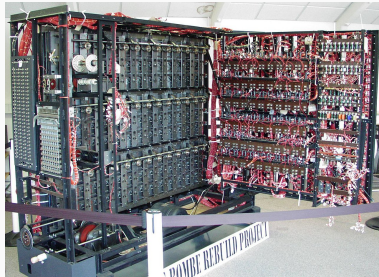
$i \leftarrow i + 1$

Retourner : La valeur de i

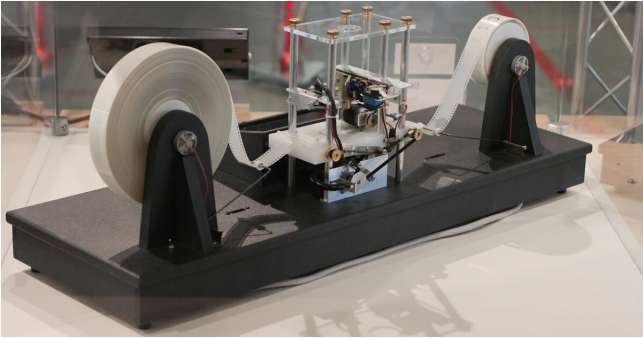
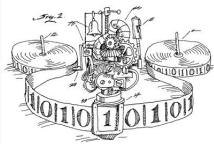
⇒ Certains programmes ne terminent pas : comment le détecter ?

Calculabilité et Complexité

Calculabilité : le modèle de la machine de Turing







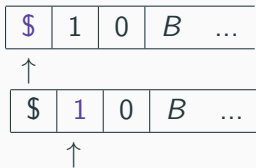
Le modèle mathématique :



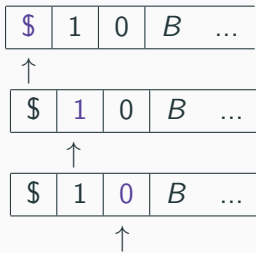
Exemple : faire +1



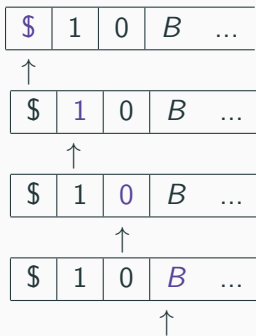
Exemple : faire +1



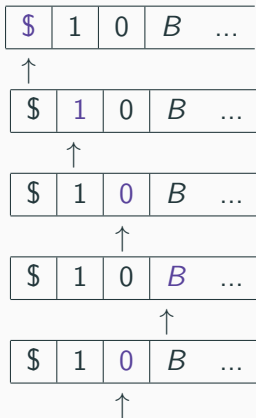
Exemple : faire +1



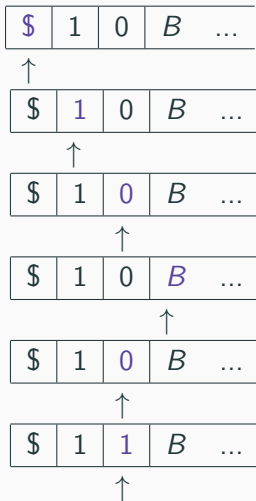
Exemple : faire +1



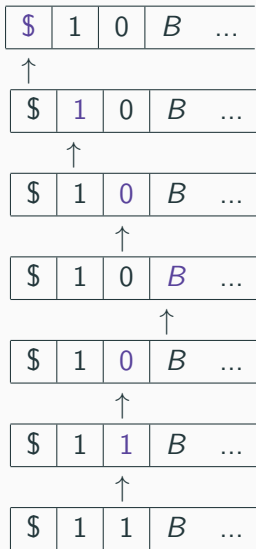
Exemple : faire +1



Exemple : faire +1



Exemple : faire +1



- Proche de ce qu'il se passe en machine
- Le langage naturel est ambigu
- Permet de faire des preuves plus facilement

Qu'est ce qu'on peut faire, ou pas ?

- Ajouter 1 à un entier ?

Qu'est ce qu'on peut faire, ou pas ?

- Ajouter 1 à un entier ? Oui !
- Savoir si un algorithme s'arrête ?

Qu'est ce qu'on peut faire, ou pas ?

- Ajouter 1 à un entier ? Oui !
- Savoir si un algorithme s'arrête ? Non...

Une amélioration d'écriture



Kateryna Lohvynivna Yushchenko

⇒ invention d'un langage de programmation "haut niveau" : traduction pour une meilleure communication humain/machine

Vers la complexité : apprendre à ranger

Vers la complexité : apprendre à ranger

