Groupes

La disparité de l'enseignement de l'informatique en classes préparatoires nous a fait adopter depuis 8 ans des groupes de niveau.

- **Forts, 2 groupes**
- **Moyens Forts, 4 groupes**
- **Moyens, 8 groupes**
- **Moyens Débutants, 4 groupes**
- **Débutants, 2 groupes**

Fonction de répartition $= 10F + 15\phi + Q + N$
$F =$ **force déclarée,** $\phi = 1$ **si fille,** 0 **sinon,**
$Q =$ **questionnaire,** $N =$ **note au concours d'entrée.**

L'informatique à l'X

Jean-Jacques.Levy@inria.fr

http://para.inria.fr/~levy

INRIA – Rocquencourt

tel: 01-39-63-56-89

Catherine Bensoussan

cb@lix.polytechnique.fr

Aile 00, LIX

tel: 54-67

http://w3.edu.polytechnique.fr/informatique/

Plan

1. L'informatique scientifique
2. Une technologie exponentielle
3. Enseignement de l'informatique à l'X
4. Le Tronc Commun d'informatique
5. Itération / Récursivité

Panthéon

AltaVista

AltaVista est le plus grand index du Web (avec HotBot)
~ 50 **millions** de requêtes par jour (de semaine)

Interface: 3 AlphaStation 500/500s, 1 GO mémoire, 6 GO disques

Moteur de recherche: 16 AlphaServer 8400 5/440' avec 12 processeurs (chacune), 8 GO ($\simeq 500$ stations personnelles modernes avec 16 MO chacune)
Index de 200 GO, adressable sur 64 bits

Connectique: ligne à 100 M bits/sec (la plus rapide privée)

Scooter: 6 M pages visitées par jour
AlphaServer 4100 5/300, 1.5 GO, 30 GO disques RAID

indexeur Web: AlphaServer 4100 5/300, 2 processeurs, 2 GO, 180 GO disque RAID.

indexeur News: AlphaServer 4100 5/440, 2 processeurs, 1 GO, 13 GO de disques.

serveur News: AlphaServer 4100 5/440 with two processors, 1 GO, 24 GO de disques.

AltaVista – bis

Comment fait-on??

Recherche en table
hachage
gestion de la mémoire secondaire
compression
parallélisme
analyse de langues naturelles
etc.

Louis Monier, Mike Burrows,
François Bourdoncle, Patrice Bertin

Ariane 501

Le lanceur a commencé à se désintégrer à environ H0 + 39 secondes sous l'effet de charges aérodynamiques dues à un angle d'attaque de plus de 20 degrés.

cet angle d'attaque avait pour origine le braquage en butée des tuyères des moteurs à propergols solides et du moteur principal Vulcain ;

le braquage des tuyères a été commandé par le logiciel du calculateur de bord (OBC) agissant sur la base des données transmises par le système de référence inertielle actif (SRI2). A cet instant, une partie de ces données ne contenait pas des données de vol proprement dites mais affichait un profil de bit spécifique de la panne du calculateur du SRI 2 qui a été interprété comme étant des données de vol ;

la raison pour laquelle le SRI 2 actif n'a pas transmis des données d'attitude correctes tient au fait que l'unité avait déclaré une panne due à une exception logiciel ;

l'OBC n'a pas pu basculer sur le SRI 1 de secours car cette unité avait déjà cessé de fonctionner durant le précédent cycle de données (période de 72 millisecondes) pour la même raison que le SRI 2 ;

l'exception logiciel interne du SRI s'est produite pendant une conversion de données de représentation flottante à 64 bits en valeurs entières à 16 bits. Le nombre en représentation flottante qui a été converti avait une valeur qui était supérieure à ce que pouvait exprimer un nombre entier à 16 bits. Il en est résulté une erreur d'opérande. Les instructions de conversion de données (en code Ada) n'étaient pas protégées contre le déclenchement d'une erreur d'opérande bien que d'autres conversions de variables comparables présentes à la même place dans le code aient été protégées ;

l'erreur s'est produite dans une partie du logiciel qui n'assure que l'alignement de la plateforme inertielle à composants liés. Ce module de logiciel calcule des résultats significatifs avant le décollage seulement. Dès que le lanceur décolle, cette fonction n'est plus d'aucune utilité ;

La fonction d'alignement est active pendant 50 secondes après le démarrage du mode vol des SRI qui se produit à H0 - 3 secondes pour Ariane 5. En conséquence, lorsque le décollage a eu lieu, cette fonction se poursuit pendant environ 40 secondes de vol. Cette séquence est une exigence Ariane 4 mais n'est pas demandé sur Ariane 5 ;

l'erreur d'opérande s'est produite sous l'effet d'une valeur élevée non prévue d'un résultat de la fonction d'alignement interne appelé BH (Biais Horizontal) et lié à la vitesse horizontale détectée par la plate-forme. Le calcul de cette valeur sert d'indicateur pour la précision de l'alignement en fonction du temps ;

la valeur BH était nettement plus élevée que la valeur escomptée car la première partie de la trajectoire d'Ariane 5 diffère de celle d'Ariane 4, ce qui se traduit par des valeurs de vitesse horizontale considérablement supérieures.

Ariane 501

Les spécifications formelles et les tests exhaustifs ont du bon.

Ariane 502

Alain Deutsch a fait l'analyse statique du code embarqué:

- accès concurrents aux variables partagées
- débordements de valeurs scalaires ou tableaux.

(200 000 lignes d'Ada; 75 000 lignes de ML pour l'analyseur).

Georges Gonthier a « fait quelques améliorations » sur des verrous mal positionnés.

Ariane 503 ??? Voir http://www.ariane503.org

Les logiciels sont des objets complexes

(surtout avec concurrence)

L'an 2000

Si `x` et `y` sont deux variables désignant des dates, l'instruction

```
if x > y then
   Faire quelquechose
else
   Faire autrechose
```

peut avoir un résultat bizarre autour de l'an 2000, car pour des raisons « d'efficacité » on a travaillé dans $\mathbf{Z}/100$

Problème: comment retrouver les endroits critiques dans des millions de lignes (Cobol)?

[cf. AnnoDomini de Henglein & Tofte (DIKU)]

Martin Jourdan travaille aussi sur ce problème à MetaWare technologies.

Quelques résultats scientifiques

1. F9 n'est pas premier
   (algorithmes et calcul distribué)
2. 20 ans de Delaunay en 1h (calcul formel)
3. Puces correctes (démonstration automatique)
4. Scanner et imagerie médicale (vision)
5. Deep Blue $>$ Kasparov (hardware)
6. Météo, physique des hautes énergies
   (super calculateurs)
7. Génôme (BD, algorithmes, calcul parallèle)
8. Cinéma (synthèse d'image)

F9 (Mark S. Manasse, Arjen Lenstra)

$2^{512}+1 =$
1340780792994259709957402499820584612747936582059239337772356144372176403007354697680187429816690342769003185818648605085375388281194656994643364900608409 $=$ 2424833
$\times$
7455602825647884208337395736200454918783366342657
$\times$
741640062627530801524787141901937474059940781097519023905821316144415759504705008092818711693940737

Avril à Juin 1990 avec

Dean Alvis, Thomas G. Arnold, Bob Ayers, Pete Balkus, Jœl Bartlett, David M. Bernardo, Anita Borg, Wieb Bosma, Patrick Boyle, Richard P. Brent, John Brownie, Bruce Bullis, Jerome Chailloux, Chran-Ham Chang, Henri Cohen, Leisa Condie, Bob Devine, Jeff Dike, John Dillon, Jeremy Dion, Mike Doody, Alan Eustace, Mike Ferrara, John Forecast, Hania Gajewska, Stephen Gildea, Andy Goldstein, Tim Greenwood, Ramsey Haddad, Gary Harding,

B. J. Herbison, Jim Horning, Felix S Hsu, Scott Huddleston, Peter Ilieve, Jeff Janock, Mike Johnson, Kevin Jones, Bill Kalsow, Irving Kaplansky, Harsh Kapoor, Bill Katz, Christopher A. Kent, Alan Kirby, Manfred Kœthe, Ted Lemon, Walter Liœn, Todd Little, Joseph A. Martin, Murray S. Mazer, James T. McCartney, Jœl McCormack, Steven Miller, Jeffrey Mogul, Francois Morain, David Mostardi, A. E. Mossberg, Jeff E. Nelson, Marc Nozell, Tom Patterson, Nigel Poole, Eric D. Postpischil, Dennis Racca, Jon Reeves, Brian Reid, Herman te Riele, John Riley, John Robinson, Eduardo Santiago, Richard Schedler, Michael Sclafani, Mark Shand, Bob Silverman, Al Simons, Kiran Somalwar, Bill Sommerfeld, Bob Souza, Alan Stanier, John T. Stapleton, Jorge Stolfi, Geof Stone, Richard Swan, Ed Taranto, Bob Thomas, Gary Thomas, Dennis Ting, Win Treese, Percy Tzelnic, Brick Verser, Paul Vixie, David Wall, Dick Wilkins, Dik Winter, Ted Wobber, Frank M. Woodall, Jr.

Calcul fait

- début mi-février 1990 à Bellcore sur 35 machines (Ds3100 et Sparc)
- mi-mars: réseau de Firefly de SRC (triplement de la puissance totale)
- fin mars: aide du réseau Internet (700 Ds3100, pour un total de 340 années de Mips)

Factorisation – bis

Plus grand nombre factorisé: $(12^{151} - 1)/11$ avec 162 chiffres décimaux.

Plus grand nombre ordinaire factorisé: RSA 130, avec quelques sites. (Arjen K. Lenstra, Bellcore, 11 avril 1996), $11527.00

RSA-130 $=$
1807082088687404805951656164405905566278102516769401349170127021450056662540244048387341127590812303371781887966563182013214880557

055 $=$
396859994595974542901611261628837860675764491128100648325551572
43

$\times$

45534498646735972188403686897274408864356301263205069600999044599

Factorisation – bis

RSA-129 factorisé en 5000 mips années, 8 mois, 600 volontaires de 20 pays. Matrice 569466×524338 **réduite à une matrice dense** 188614×188160 **par élimination gaussienne, 4.13 GO, 45h de MasPar MP-1 16K. $100.00**

Factorisation de C120: 61 machines (Richard Wackerbarth, 5 mars 1996) avec l'aide de
`Damien.Doligez@inria.fr`, `HJSmith@ix.netcom.com`, `Michel.Kern@inria.fr`, `Paul.Zimmermann@loria.fr`, `etc`

Factorisation – ter [E. Mayer et F. Morain.]

MC7331$= (2^{7331} - 1)/458072843161$ **est premier**

=15486870514843934124155218820066836943673540605667846014406085245674
3150967504690978407922763663102772174710356841717193060814649307018188
5684937145152644900693138604087142686641512220288382498825157526200289
1014031422811450801496740052595645022554644043734181593867741102982433
4161228294335639550029715077499809109707953037454360247234092179489604
9052372911232513806875984433419377119508372179809146417629299856435841
7271267183842978433675018896209477564122847545295022273605303157929654
7823316839675761753847917853260032220931844178707732900657707079865988
7711632535399679553348723682652716524087315530167665366573728619034852
7508918872890080234563651127033145399631564739631343520044434872941033
5802041143218488857347212894294071614635331134440778476355549492605208
9966862574381858439749025896147532210155747182211641285533074807579946
8342388288180426165596672796011755991342622265320903007353907181294078
4146048197708602871314543477676254216646348341108732566943428915883071
1395219892406794005146541570568903308842052488621169209173350448362618
8643126441652575936332708732674211736866544795337878439808480369012274
78356435540415966187604983802612377740444013537775997244926584931086571
1411788778796032582771544936514666629531569179292105302336412796770354

8995887018805152638975520311684748706264045079484060231842396893729447
5058165123576504111272244568835982190532825940368488448085275252279252
6350561020196480481650104630749988910714715688254327480880453082835592
0841730365612394746540601952468801892495523026521836562398801075510110
5688528218793277235966284900404256341457629096767363722062203067330746
9490035273111451650441198662298038956839643432726691739614518315990382
9325952085282159087208405487745859121867472781850508401067173985213817
6399690340741882803008297306029931684623931702424310030319385165818895
8452960807841688830455597022824855831313338581039055206954623357516677
9003085172716228614376595246017420905612464208768664379009237348918600
9462055798303939299506487626919962545659356318475346262638660772213385
7088919174178268237118111988794504533306145503511742537418916935519997
8480934668594102981332702586813536001837675718675943723107772772557420
5887656324617343711044714327

(un mois de CPU sur une alpha 400 MHz)

Démonstration automatique ou assistée

Matériel

vérification d'un protocole de communications entre les caches de la machine multi-processeurs GigaMax, Encore. *McMillan* (CMU), 1992

idem pour un chip de communications déja réalisé et qui ne fonctionnait pas. *Fujita*, Fujitsu, 1993

conception hiérarchique du Power PC, IBM, 1992 ($\simeq$ vieille méthode Bull). Aussi chez DEC.

automates de Buchi pour spécifier et vérifier un circuit de gestion de communications. *Kurshan*, (ATT Bell Labs)

Démonstration automatique ou assistée

Pour le logiciel, c'est plus dur

preuves des programmes concurrents

preuves des gros programmes . . .

preuves de protocoles de communication: "Bounded Retransmission Protocol", (Philips Research), en 1994.

axiomatisation en TLA de protocole de contrôle de flux pour des lignes série très rapide, (Bull recherche), 1994.

protocole banquaire C-SET (*Chip-Secure Electronic Transactions*), 1997.

cartes banquaires téléchargeables, Java Card, 1998.

Les domaines

langages de programmation	**algorithmique**
validation des programmes	**algorithmique géométrique**
démonstration automatique	**algorithmique parallèle**
parallélisme et communication	**analyse d'algorithmes**
logique mathématique	**codes et cryptographie**
langages formels et compilation	*zero knowledge algorithms*
bases de donnée	**calcul formel**
systèmes d'exploitation	**distribution/mobilité**
matériel et architecture	**réseaux**
CAO	**multimedia**
vision	**interfaces**
robotique	**systèmes réactifs**
...	

L'informatique traite surtout de calculs sur des structures de données autres que numériques.

Une technologie exponentielle

	gain	période
processeur vitesse	$\times 1.35$	1 an
capacité mémoire	$\times 4$	3 ans
coût/octet disques	$\times 0.75$	1 an

En 1998,

Processeur à 600 MHz, 10M transistors, ($\simeq 1300 \times$ Vax)
RAM 64 Mbits, 100 MHz
Disques 2.5" à 8GO, 3.5" à 18GO (10 kF)
Réseaux ATM à 640 M bits/sec
Internet avec 37 M machines

Le cyber-espace — défi des années 90

- consultation de bibliothèques
- minitel planétaire
- commerce électronique
- courrier électronique
- échanges de programmes
- sécurité et confidentialité
- téléphonie et audio
- télévision et vidéo à la demande
- réglementations

Réseaux + informatique = grand public

Evolution des systèmes informatiques

66: GE-645	mémoire virtuelle, Multics
70: pdp11	Unix
72: alto	bitmap, souris, wysiwyg
74: ethernet	réseau entre les altos
75: dover	impression laser, Press
78: Vax et Unix-BSD	Unix avec mémoire virtuelle
82: Sun Microsystems	Station personnelle Unix
84: uucp	Interconnexion à longue distance (IP)
88: Machines RISC	IBM 801, Titan, SPARC+Mips
91: Portables, format A4	Poids et autonomie
93: Mosaic, Internet	Grand public

Quelques noms

Organick	Multics	MIT
Thompson,Ritchie	Unix	Bell labs
Lampson,Thacker	Alto	Xerox-PARC
Metcalfe	Ethernet	Xerox-PARC
Warnock	Postscript	Xerox-PARC
Bell	pdp11, Vax	CMU
Joy	Unix BSD, SunOS	UCB
Leffler	Unix + réseau	UCB
Hennessy, Patterson	RISC, Mips	Stanford/UCB
Kay	portables, OO	Xerox-PARC
Berners-Lee	Html	CERN

Le *dynabook* d'Alan Kay

"The Learning Research Group at Xerox PARC is concerned with all aspects of the communication and manipulation of knowledge. We design, build, and use dynamic media which can be used by human beings of all ages. Several years ago, we crystallized our dreams into a design idea for a personal dynamic medium the size of a notebook (the Dynabook) which could be owned by everyone and could have the power to handle virtually all of its owner's information-related needs..."

Alan Kay et Adele Goldberg,

Xerox Palo Alto Research Center,

Computers, Mars 1977

L'informatique et l'X

Futurs ingénieurs doivent connaître l'informatique

Former de bons informaticiens (vs ENS, ETH, MIT)

Format du cours demande un peu de travail personnel (beaucoup moins que les 15h/semaine du MIT ou autres)

Formation initiale très hétérogène des élèves (encore pire en 98)

Mathématiques $\Rightarrow$ rigueur $\Rightarrow$ atout en informatique (vrai en général)

Recherche en informatique

Le matériel informatique du Tronc Commun

2 serveurs fichiers, 2 serveurs communication

$\sim$ 100 postes de travail, [10 Sun, 2 SGI, 24 PowerMac, 24 PC Linux, 17 Hp 9000, 10 dec/alpha stations, 12 HP b132, 31 tX]

réseau *up-to-date* dans les chambres de l'X

abandon des portables en 98

L'environnement informatique est bon. Il faut s'en servir.

Quelques recommandations

Sécurité réseau

CERT (*Computer Emergency Response Team*), dépend de la NSA (*National Security Agency*), fait la police sur le réseau.

Le réseau n'est opérationnel que si une auto discipline (simple) est respectée.

Centre Informatique de l'X

Pas des *punching balls*. Climat de confiance. Très difficile à 460 (900). Si pbs, on coupe l'Internet.

Impression ($\simeq$ 10-20 centimes la page). Ne pas faire des impressions inutiles.
Utiliser les photocopieuses.

Elèves

Un délégué par PC
Repas ensemble à la fin novembre
2 délégués info pour la promo

Enseignants

Ecole polytechnique: 1 prof, 10 mdc , 12 ctp, 11 vacataires, 1 secrétaire.

Cours: 4 prof univ., 6 DR, 1 IA

TD: 1 DR, 9 CR, 2 MdC, 2 ingénieurs, 9 thésards.

Utiliser les pages Web et le courrier électronique

Enseignement de l'informatique à l'X

1. Tronc Commun, obligatoire
   1er trimestre, 1ère année;
   Projet informatique,
   2ème semestre, 1ère année, ou
   1er trimestre, 2ème année

2. Voie C, optionnelle
   2ème semestres, 1ère année
   Optimisation combinatoire

3. Majeure 1, optionnelle
   1er trimestre, 2ème année

4. Majeure 2, optionnelle
   2ème trimestre, 2ème année

Contenus

Tronc Commun: algorithmes et programmation
Lévy (et Cori)

Voie C: optimisation combinatoire
Steyaert

Majeure 1: mathématiques et informatique
Vuillemin
algèbre des codes, architecture, logique,
vision et synthèse, complexité,
programmation objet en Java

Majeure 2: informatique
Cori (et Lévy)
compilation, systèmes et réseaux, bases de données,
algorithmique NP, géométrie [Chazelle], calcul parallèle

Plan du TC

1. 3 initiations: mardi 29 Sep – mardi 3 Nov.
2. 10 petites classes: jeudi 5 Nov – jeudi 28 Jan.
3. 10 travaux dirigés: lundi 16 Nov – mardi 2 Fév.
4. contrôle HC 2h: mercredi 16 Déc.
5. composition de classement 4h: lundi 15 Fév,
   Cœfficient 10.
   Théorique.
6. projets de programmation, 2ème semestre, ou 2ème année
   amphi de présentation: xx Déc?
   500–1000 lignes de Caml, Java, C ou etc,
   ~20 sujets différents de difficultés variées,
   Cœfficient 8.
   Pratique, travail personnel.
   Souvent pluridisciplinaire.

Le contenu du Tronc Commun

1. Structures de données élémentaires (tableaux, listes, files)
2. Arbres
3. Graphes
4. Récursivité
5. Modularité
6. Analyse syntaxique élémentaire

Langages de programmation

Java, Gosling et Sun microsystems

ML / Caml, Milner et INRIA

Les langages de programmation du Tronc Commun

Fortement typés
($\Rightarrow$ jamais *core dumped*, **ni** *segment violation fault***)**

Gestion automatique de la mémoire
($\Rightarrow$ pas à se préoccuper de la taille des objets)

Ce sont des langages modernes

Java, $\simeq$ le mieux à attendre de la communauté C ou C++,
peut-être un problème d'efficacité?

ML, avec une écriture fonctionelle/algébrique et des types sophistiqués,
Objective Caml (Ocaml) est très efficace.

Java nouveau en 98 au Tronc Commun

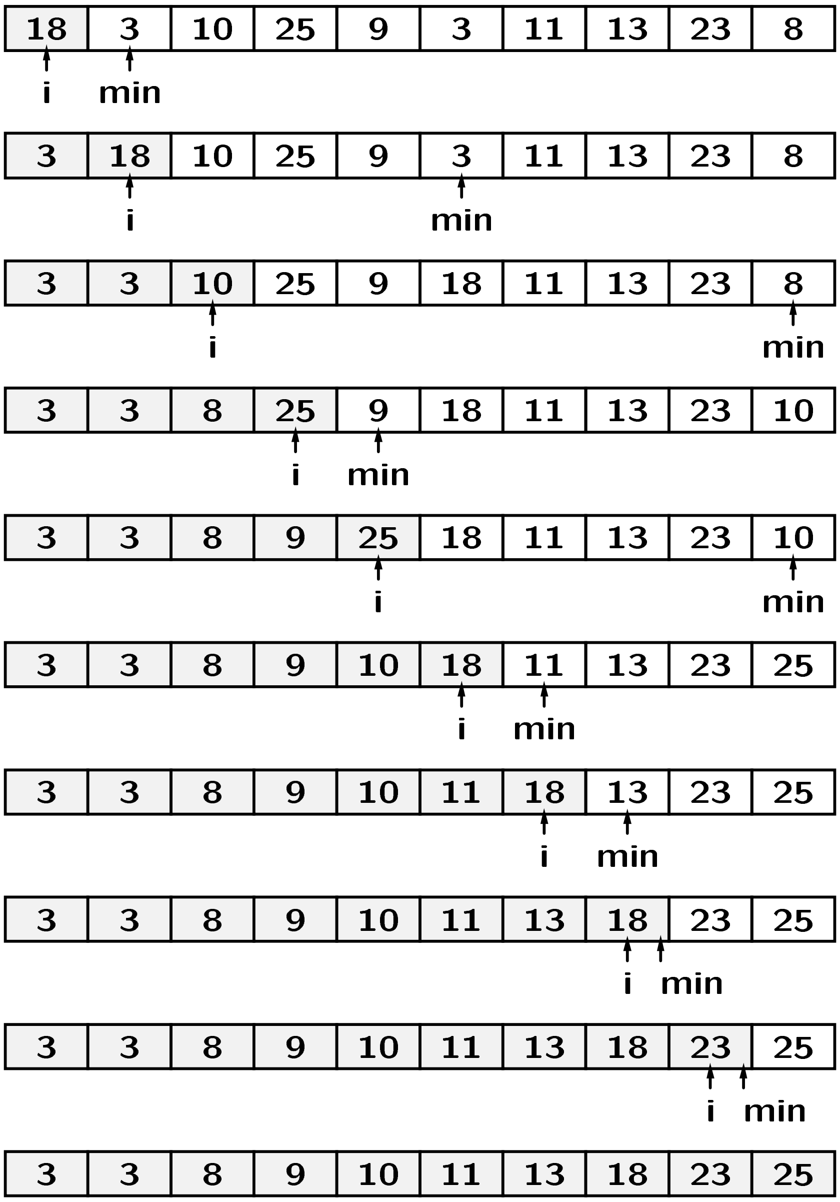
Programmation

1. Itération —— Tableaux, Listes
2. Récursivité —— Arbres, Graphes
3. Modularité – Gros programmes
4. Exploration – Arbre de recherche
5. Analyse syntaxique
   Chaînes de caractères $\Longrightarrow$ Arbres
6. Portabilité

Maxime du TC

Le matériel passe, le logiciel reste

Le tri par sélection (programme itératif)



Le tri par sélection (bis)

Algorithme

Pour $i = 1$ **à** N **faire**

> **Chercher l'emplacement** min **de l'élément minimum dans** $[i+1, N]$**,**
> **Echanger** $a[i]$ **et** $a[min]$**.**

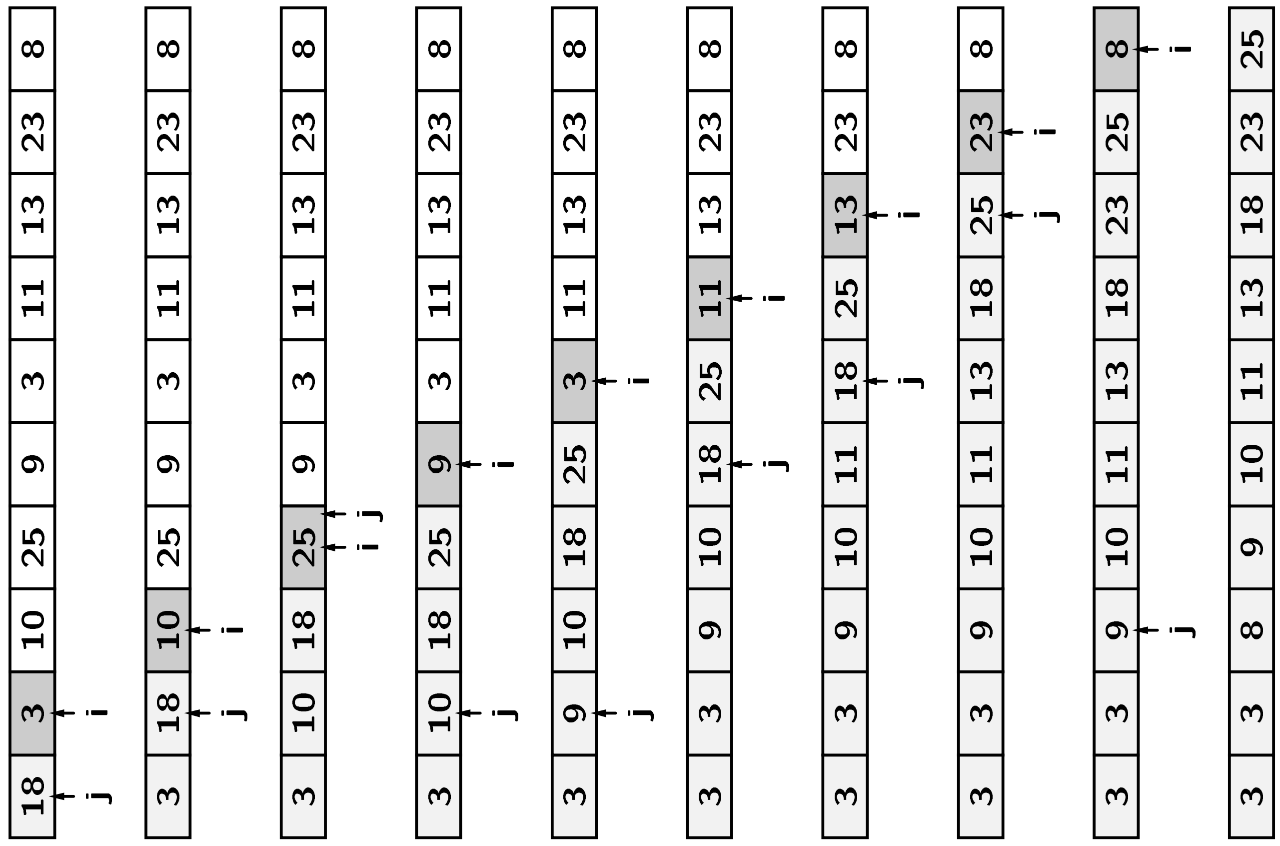
Correction

> **Début:** $\langle a[1] \rangle$ **trié**
>
> **A tout moment,** $\langle a[1], a[2], \cdots, a[i] \rangle$ **triés**
>
> **A la fin,** $i = N$ **et** $\langle a[1], a[2], \cdots, a[N] \rangle$ **triés**

Coût

> $N - 1 + N - 2 + \cdots + 1$ **comparaisons**
>
> $= N(N-1)/2$
>
> $\simeq O(N^2)$ **opérations**

Le tri par insertion



Le tri par insertion

Algorithme

Pour $i = 2$ **à** N **faire**

> **A partir de** i**, aller vers la gauche et trouver l'emplacement où insérer** $a[i]$ **dans** $[1, i-1]$
> **Insérer** $a[i]$ **en décalant vers la droite les éléments visités.**

Correction

> **Début:** $\langle a[1] \rangle$ **trié**
>
> **A tout moment,** $\langle a[1], a[2], \cdots, a[i-1] \rangle$ **triés**
>
> **A la fin,** $i = N$ **et** $\langle a[1], a[2], \cdots, a[N-1] \rangle$ **triés et après l'insertion de** $a[N]$**, on a** $\langle a[1], a[2], \cdots, a[N] \rangle$ **triés**

Coût

> $1 + 2 + \cdots + N - 1$ **comparaisons**
> $= N(N-1)/2$
> $\simeq O(N^2)$ **opérations**

Le tri par fusion (programme récursif)

18	3	10	25	9	3	11	13	23	8
3	18	10	9	25	3	11	13	8	23
3	18	9	10	25	3	11	8	13	23
3	9	10	18	25	3	8	11	13	23
3	3	8	9	10	11	13	18	23	25

Le tri par fusion (programme récursif)

Algorithme

Si $N = 0$ (tableau vide), ne rien faire.

Sinon:
1- Trier la moitié gauche du tableau a.
2- Trier la moitié droite du tableau a.
3- Fusionner les 2 résultats.

Correction

immédiate.

Coût

$$C_N = 2C_{\lfloor N/2 \rfloor} + N$$

$$C_0 = 0$$

$$C_N \simeq N \log N$$

Le gain vient de la décomposition récursive du tableau a.

Les Arbres (structure de données récursive)

Définition

Arbre $=$ {**Vide**} $\uplus$ **Arbre** $\times$ **Element** $\times$ **Arbre**

Exemple d'algorithme sur les arbres

Taille (Vide) $=$ **0**

Taille (Arbre(g, n, d))

$=$ **1** $+$ **Taille (g)** $+$ **Taille (d)**

Correction

immédiate.

Coût (N nœuds)

$C_N = N$

On parcourt un arbre récursivement en temps linéaire.

Itération / Récursion

- Programmes récursifs et structures de données récursives se correspondent.
- L'écriture récursive est presque toujours plus élégante.
- Induction mathématique et récursivité se correspondent souvent.
- La difficulté est toujours de trouver les structures récursives associées à des problèmes combinatoires (exemple: test de planarité des graphes, *depth first search*, Tarjan 72).
- Tous les langages modernes autorisent la récursivité.
- Certains permettent aussi l'écriture récursive des structures de données (par exemple Caml)