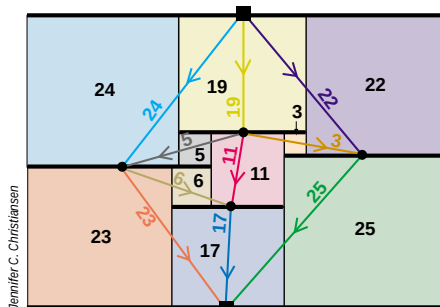




2. Un rectangle carré peut être représenté par un circuit électrique. Chaque ligne horizontale correspond à un nœud. Le courant entre par le nœud supérieur et sort par le nœud inférieur. Son intensité, sur chaque fil conducteur (ou arête du graphe) est égale à la longueur du côté du carré.

mère joua avec...et reconstitua le rectangle, mais sa solution différait de celle de son fils. C'était la première fois que l'on trouvait ainsi un même rectangle quadrable de deux façons différentes.

On avait espéré trouver deux rectangles de même dimension, quadrables par des pavés différents, car, en ajoutant deux carrés, on aurait obtenu une quadrature de carré. Le rectangle de R. Brooks ne conduisait pas à une telle solution, car le même ensemble de pavés est utilisé dans les deux solutions. Toutefois, en examinant les deux diagrammes de Smith des deux pavages, les mathématiciens comprirent qu'en court-circuitant deux sommets d'un diagramme, ils obtenaient l'autre diagramme. Les deux points ainsi «identifiés» (le terme mathématique qui correspond au court-circuit) étaient au même potentiel électrique, de sorte que les courants électriques n'étaient pas modifiés.

Partant de cette observation, ils trouvèrent d'autres moyens d'utiliser les diagrammes de Smith et, finalement, trouvèrent un carré simplement quadrable par 69 pavés. R. Brooks trouva ensuite un carré simplement quadrable par seulement 39 pavés.

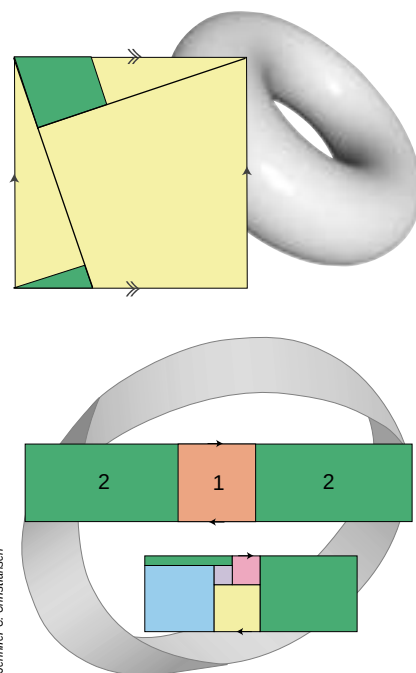
En 1948, T. Wilcocks trouva un carré quadrable par 24 pavés, mais le pavage n'était pas simple. Vers la même époque, J. Bouwkamp et ses collègues cataloguèrent tous les rectangles quadrables avec moins de 15 pavés : il en existe 3 663. En 1962, A. Duivestijn démontra que tout carré simplement quadrable doit contenir au moins 21 pavés ; puis, vers 1978, il trouva un tel carré et prouva qu'il était le seul à 21 pavés. En 1992, J. Bouwkamp et A. Duivestijn publièrent les 207 carrés simplement quadrables avec 21 à 25 pavés : leur liste est exhaustive.

Bien que ces résultats résolvent presque la quadrature du carré, bien des variantes du problème demeurent ouvertes. Comment quadrer les dominos, ou rectangles deux fois plus longs que larges ?

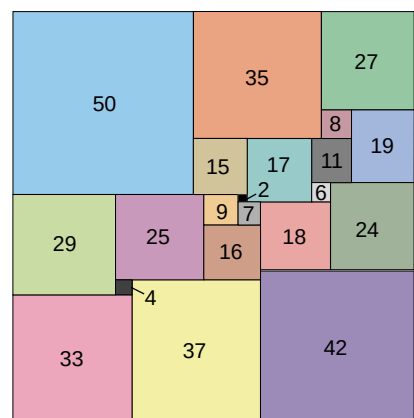
Il existe une quadrature évidente, avec deux carrés, mais en existe-t-il d'autres ?

Une autre possibilité consiste à paver d'autres surfaces que des rectangles : c'est ce qu'a étudié David Gale. Les topologistes savent que des surfaces intéressantes apparaissent quand on recolle (que l'on «identifie») les côtés opposés d'un rectangle, avec ou sans torsion. Le recollement sans torsion des deux premiers côtés d'un rectangle conduit à un cylindre. Si l'on recolle deux côtés opposés après avoir tordu le bord d'un demi-tour, on forme un ruban de Möbius (voir la figure 3). Quand on recolle les quatre côtés par paire de côtés opposés et sans torsion, on obtient un tore. Quand une paire de côtés est recollée après une torsion d'un demi-tour et que l'autre paire est recollée sans torsion, on obtient une bouteille de Klein, sans intérieur ni extérieur. Enfin, quand on recolle les côtés deux à deux, en tordant d'un demi-tour dans les deux cas, on obtient un plan projectif, impossible à représenter dans notre espace à trois dimensions.

Évidemment tout pavage du rectangle conduit à un pavage de la surface formée après un de ces recollements, mais cette surface peut posséder d'autres pavages que ceux du rectangle initial, parce que ses pavés peuvent déborder des bords du rectangle. La partie inférieure de la figure 3 montre un ruban de Möbius pavé avec deux carrés seulement, de côtés respectivement égaux à 1 et à 2. Les flèches indiquent les recollements. Bien



3. Tore (en haut) et ruban de Möbius (en bas) peuvent être pavés. Les côtés marqués de flèches similaires doivent être recollés, de sorte que les flèches coïncident. Les zones vertes forment alors un pavé, de même que les zones jaunes.



4. L'unique carré quadré par le nombre minimal de pavés : 21.

qu'un carré semble être coupé en deux rectangles, ces deux parties sont réunies lorsque les côtés sont identifiés. Cependant le pavage du ruban de Möbius a une propriété désagréable : le petit carré a une frontière commune avec lui-même. Deux de ses côtés opposés sont collés ensemble, de sorte qu'il devient lui-même un ruban de Möbius, et non un carré. En 1993, S. Chapman découvrit un pavage du ruban de Möbius sans cette particularité, avec cinq pavés (bande inférieure de la figure 3). Aucun pavage ne comporte moins de pavés.

Un cylindre peut également être quadré, mais avec au moins neuf pavés, comme pour les rectangles. Des pavages «triviaux» proviennent simplement des rectangles de Moron, avec un recollement des deux côtés, mais il existe également d'autres pavages non triviaux à neuf pavés. Ces derniers ont la même taille que les pavés de Moron.

Pour le cylindre comme pour le ruban de Möbius, les côtés des carrés doivent être parallèles aux bords des surfaces. Cependant ni le tore, ni la bouteille de Klein, ni le plan projectif n'ont de bords, de sorte que les pavés peuvent être également en biais. De fait, on peut ainsi paver un tore avec seulement deux carrés, pourvu que l'on accepte que deux côtés d'un même carré se touchent (en prime, une preuve du théorème de Pythagore est cachée dans cette figure : la voyez-vous ?).

On connaît mal le pavage d'une bouteille de Klein. Tout pavage d'un ruban de Möbius donne celui de la bouteille de Klein par recollement du ruban le long de son unique bord, et il n'existe pas d'autre manière de paver une telle bouteille avec au plus six pavés carrés. On ignore si la propriété reste vraie pour sept ou huit pavés, et l'on sait qu'elle est fautive pour neuf pavés.

Enfin le pavage du plan projectif est un problème ouvert, tout comme le pavage du cube : à vous !



L'art du tri

JEAN-PAUL DELAHAYE

La loi du moindre effort dans les méthodes de tri de données ou de cartes. Gagnez un temps précieux !

Pour vérifier qu'un jeu est complet ou identifier les cartes absentes, pour préparer certains tours de prestidigitation ou un tournoi de bridge, il faut trier les cartes. Les problèmes posés par ces tris ressemblent à ceux que doivent traiter les informaticiens pour classer des données. Pour les résoudre, une science inattendue de l'économie des gestes et du travail a été mise en œuvre.

Les mathématiciens ont compris il y a 40 ans que trier n'était pas si simple et, depuis, ils ont étudié comment préserver le précieux temps de calcul des ordinateurs. Parmi des dizaines de méthodes proposées, certains algorithmes remarquables de tri informatique s'adaptent au classement des cartes et illustrent que le plus rapide n'est pas le plus simple. Mais, comme on le verra aussi, le classement des cartes a ses propres subtilités.

LE TRI PAR INSERTION

Le tri *par insertion* est sans doute le plus naturel de tous les tris à la main, et c'est d'ailleurs celui que la plupart des gens utilisent pour classer leurs cartes quand ils ramassent leur main au bridge ou quand ils doivent trier alphabétiquement des noms, des fiches, ou des livres.

L'algorithme de tri par insertion est le suivant : on prend les cartes une à une, et l'on compose un nouveau paquet en plaçant chaque carte qu'on introduit à la place qu'elle doit avoir parmi celles qui sont déjà placées. À chaque instant du tri par insertion, une partie du paquet est parfaitement classée et l'autre encore désordonnée.

On mesure la complexité d'un tri en comptant le nombre de comparaisons entre deux cartes (ou deux nombres s'il s'agit de nombres) qu'on doit faire pour classer l'ensemble. Plus ce nombre est grand, moins bon est le tri ; plus il est petit, meilleur il est.

Étudions ce qui se passe pour le tri par insertion. Quand on place la première carte, on n'effectue aucune comparaison. Quand on place la seconde carte, on effectue une comparaison : il faut savoir si la carte qu'on introduit est plus faible ou plus forte que celle qui est déjà placée. Quand on place la troisième carte, on effectue deux comparaisons : la carte nouvelle est comparée avec chacune des deux cartes déjà en place pour savoir si elle doit se placer devant les deux, au milieu ou derrière. Et ainsi de suite : le placement de la carte numéro p peut nécessiter $p - 1$ comparaisons. Au total donc, on aura effectué au plus $1 + 2 + \dots + (n - 1) = n(n - 1)/2$ comparaisons quand la n -ième carte aura été placée. J'ai bien précisé «au plus», car, quand vous placez la carte numéro p , vous n'avez vraiment besoin d'effectuer $p - 1$ comparaisons que si sa place est en dernier (vous pouvez placer la carte p dès que vous rencontrez une carte plus forte).

Pour 10 données, cela fait au plus 45 comparaisons ; pour 100 données, au plus 4 950 ; pour 1 000 données, au plus 499 500. Pour obtenir le nombre moyen de comparaisons, il faut diviser ces nombres par 2.

L'augmentation du travail à faire (dans le pire cas, ou en moyenne) est une fonction polynôme de degré deux du nombre de cartes. On dit que l'algorithme est quadratique et en pratique cela signifie qu'il devient rapidement inutilisable.

LE TRI-RAPIDE

Le tri-rapide est plus difficile à effectuer, mais le jeu en vaut la chandelle, car il nécessite, au total, bien moins de comparaisons. Son avantage par rapport au tri par insertion est très net quand le nombre de cartes à trier augmente. Nous décrivons cet algorithme par étapes.

Étape 1. On prend la première carte (qu'on appellera carte pivot), on la place sur la table au milieu et en haut. La carte pivot constitue la ligne 1. On forme alors deux paquets en comparant toutes les cartes restantes avec la carte pivot : les cartes plus faibles sont mises en paquet à gauche sur la ligne 2, les plus fortes sont mises à droite sur la ligne 2 (voir la figure 2).

Étape 2. On prend toutes les cartes du paquet de gauche de la ligne 2, sauf la première qui sert comme précédemment de carte pivot et qu'on laisse sur la ligne 2. On sépare comme précédemment ce paquet en deux nouveaux paquets plus petits qu'on place sur la troisième ligne. Le paquet de droite de la ligne 2 est traité de la même manière. Il y a donc maintenant une carte sur la ligne 1, deux sur la ligne 2, et quatre petits paquets sur la ligne 3.

Étape 3. On retire une carte de chaque paquet de la ligne 3 qui y reste et l'on sépare le reste de chaque paquet comme précédemment en deux, ce qui donne en tout 8 paquets qu'on place sur la ligne 4.

On continue de même jusqu'à n'avoir plus que des cartes seules, chaque paquet ayant été totalement fractionné. Les cartes sont alors disposées en une structure ressemblant à un arbre à l'envers. À chaque carte de l'arbre sur la ligne n sont associées deux cartes au plus de la ligne $n + 1$, qu'on appellera ses fils gauche et droit ; la carte sera bien sûr appelée la carte père de ses deux fils (le mouvement *Politically Correct*, aux États-Unis, oblige les informaticiens à parler de cartes filles et de cartes mères, mais en France nous pouvons garder une certaine liberté de langage).

Quand toutes les cartes sont ainsi disposées en un arbre à l'envers, on les ramasse en «repliant l'arbre» jusqu'à n'avoir plus qu'un seul paquet (qui est alors classé) : on repère un paquet sur une ligne n n'ayant aucun paquet asso-

cié sur la ligne $n + 1$ (le paquet n'a pas de fils), on le place sur le paquet correspondant (son père) de la ligne $n - 1$ s'il s'agit d'un fils gauche, et en dessous s'il s'agit d'un fils droit.

La complication de ce tri inventé en 1962 par C. Hoare en vaut vraiment la peine, car le nombre de comparaisons qu'on doit réaliser pour classer n cartes (ou données) est cette fois en moyenne à peu près $n \cdot \log_2(n)$. Dans cette formule, il faut arrondir le logarithme en base 2 de n à l'entier supérieur (rappelons que $\log_2(2) = 1$, $\log_2(4) = 2$, $\log_2(8) = 3$, etc.) En définitive, pour 10 données, il faut environ 40 comparaisons, pour 100 données, environ 700, et pour 1 000, environ 10 000, ce qui est sensiblement moins qu'avec le tri par insertion.

La justification de ce $n \cdot \log_2(n)$ est la suivante. D'une part, pour construire chaque ligne, il faut comparer chaque carte restante à une carte de la ligne précédente. Il faut donc faire environ n comparaisons à chaque construction de ligne. D'autre part, le nombre de lignes est environ $\log_2(n)$, car il y a 1 carte au niveau 1, 2 cartes au niveau 2, 4 au niveau 3, 2^{n-1} au niveau n . Ce

calcul n'est qu'une évaluation, car certains paquets s'épuisent plus vite que d'autres, mais l'idée expliquée peut être rendue parfaitement rigoureuse.

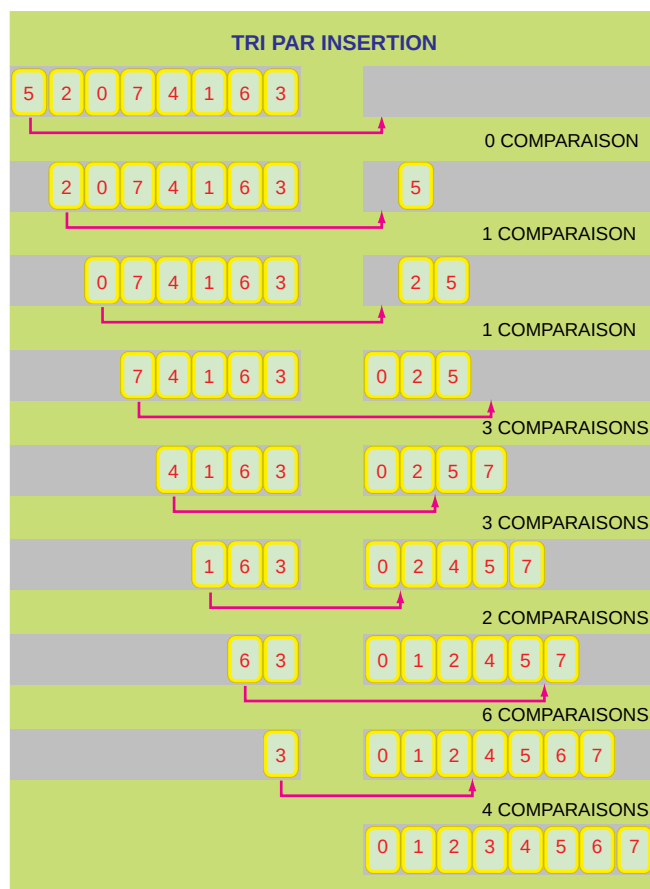
Dans les mauvais cas (quand la séparation par le pivot est systématiquement déséquilibrée), le tri-rapide a exactement une complexité de $n(n-1)/2$ comparaisons, c'est-à-dire la complexité du tri par insertion : ce qui distingue les deux tris, c'est le comportement en moyenne, qui est toujours quadratique pour le tri par insertion, alors qu'il est en $n \cdot \log_2(n)$ pour le tri-rapide.

Assez paradoxalement, les cas où le tri-rapide est le plus lent se présentent quand le paquet initial est déjà totalement ou partiellement trié. Statistiquement, ces mauvais cas se présentent très rarement (d'où la moyenne en $n \cdot \log_2(n)$). Cependant il se trouve que les données qu'on doit trier ne sont bien souvent pas statistiquement quelconques et qu'en pratique elles sont au contraire souvent partiellement ou totalement triées : se reposer sur la rareté statistique (très différente de la rareté pratique) des mauvais cas est donc un mauvais calcul (que certains programmeurs mal informés font pourtant).

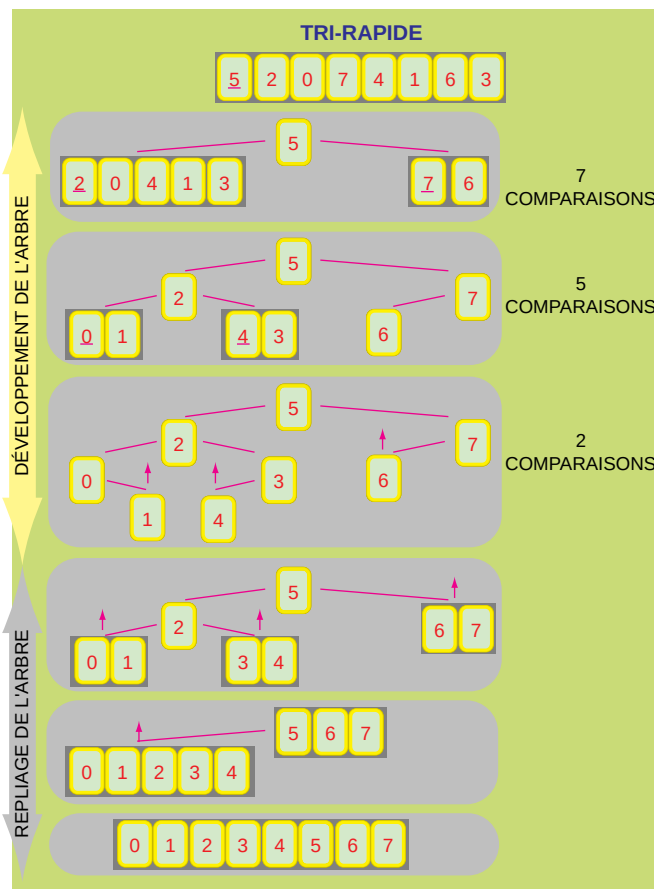
Il existe heureusement un moyen simple d'arranger les choses et qui permet de considérer en définitive que le tri-rapide est un bon algorithme : il faut mélanger les données avant de les trier ! C'est aujourd'hui ce qu'on recommande à ceux qui, en informatique, veulent utiliser le tri-rapide. Avec cette précaution, le tri-rapide est alors très efficace, et bien meilleur que le tri par insertion.

LE MIEUX QUE L'ON PUISSE FAIRE

C'est même un algorithme qu'on ne peut pas espérer améliorer, car $n \cdot \log_2(n)$ est la borne la meilleure qu'on puisse atteindre : aucun algorithme de tri de données n'utilisant que des comparaisons ne peut en faire moins de $n \cdot \log_2(n)$ en moyenne. La justification de ce résultat est intéressante. Notons qu'il y a $n!$ classements différents des cartes et que pour les trier il faut nécessairement les discriminer les uns des autres. Avec une comparaison, on ne peut discriminer que 2 classements ; avec 2 comparaisons, on ne peut discriminer que $4 = 2^2$ classements, etc. Donc, pour discriminer les $n!$ classements possibles



1. Le tri par insertion est naturel et simple à programmer. Malheureusement, ce n'est pas un tri efficace, car, pour trier n données, il lui faut en moyenne effectuer $n^2/4$ comparaisons.



2. Le tri-rapide est beaucoup plus efficace, même s'il est plus complexe à programmer : pour trier n données, il lui faut, en moyenne, effectuer $n \cdot \log_2 n$ comparaisons.

de n cartes, il faut un nombre de comparaisons m tel que $2^m \geq n!$. Le résultat annoncé s'obtient maintenant en utilisant la célèbre formule de Stirling, $n! \approx \sqrt{(2\pi n)} (n/e)^n$, qui donne $\log_2(n!) \approx n \log_2(n)$. Pour discriminer tous les classements possibles de cartes (ce qui est nécessaire pour trier les cartes), il faut donc au moins effectuer $n \cdot \log_2(n)$ comparaisons.

TRI DE CARTES PERFORÉES

Pour les cartes, le tri-rapide n'est pas agréable, il est encombrant. De plus, le tri-rapide n'utilise pas l'information que, dans un jeu normal, il y a une carte de chaque sorte ni plus, ni moins.

En informatique, bien souvent, on ne sait pas si les données qu'on doit trier appartiennent ainsi à un ensemble de valeurs réduit, mais si c'est le cas il existe d'autres algorithmes qui peuvent être préférés au tri-rapide.

Lorsqu'on doit trier des numéros dont on sait qu'ils forment une suite complète (ou seulement contenue dans un intervalle assez petit), il existe un algorithme amusant qui fut utilisé à l'époque des cartes perforées, car il

pouvait s'effectuer mécaniquement avec une grande rapidité.

Je vais présenter l'idée de ce tri appelé *tri par base* sur un exemple. Imaginons que nous voulions trier les nombres suivants, tous inférieurs à 16 : 2 ; 7 ; 4 ; 9 ; 1 ; 15 ; 3 ; 13 ; 6 ; 12 ; 5. On les écrit tous en base 2, ce qui donne cette liste de nombres binaires de quatre chiffres binaires au plus chacun : 10 ; 111 ; 100 ; 1001 ; 1 ; 1111 ; 11 ; 1101 ; 110 ; 1100 ; 101.

On les classe selon le dernier chiffre binaire : ceux dont ce dernier chiffre binaire est 0 sont amenés en tête : 10 ; 100 ; 110 ; 1100 ; 111 ; 1001 ; 1 ; 1111 ; 11 ; 1101 ; 101.

On les classe selon l'avant-dernier chiffre binaire : ceux dont cet avant-dernier chiffre binaire est 0 sont amenés en tête en respectant l'ordre qu'ils ont les uns par rapport aux autres (c'est important) :

100 ; 1100 ; 1001 ; 1 ; 1101 ; 101 ; 10 ; 110 ; 111 ; 1111 ; 11.

On continue ainsi en les classant selon l'avant-avant-dernier chiffre binaire, puis l'avant-avant-avant-dernier chiffre binaire.

1001 ; 1 ; 10 ; 11 ; 100 ; 1100 ; 1101 ; 101 ; 110 ; 111 ; 1111.

1 ; 10 ; 11 ; 100 ; 101 ; 110 ; 111 ; 1001 ; 1100 ; 1101 ; 1111.

Cette fois, c'est bon :

1 ; 2 ; 3 ; 4 ; 5 ; 6 ; 7 ; 9 ; 12 ; 13 ; 15

Avec des cartes perforées, il suffisait d'écrire sur chaque carte son numéro en notation binaire, de sélectionner mécaniquement toutes les cartes dont le dernier chiffre est un zéro, de les extraire d'un coup et de les faire passer devant, etc. Une méthode d'encoche permet (voir la figure 3) de faire l'opération de sélection instantanément avec une tige métallique.

Ce tri appartient à l'histoire de l'informatique, et d'ailleurs sa première description est due à L. J. Comrie qui, en 1929, l'a introduite en présentant des équipements pour cartes perforées. Du matériel de bureau dans les années 1960 était fondé sur cette méthode.

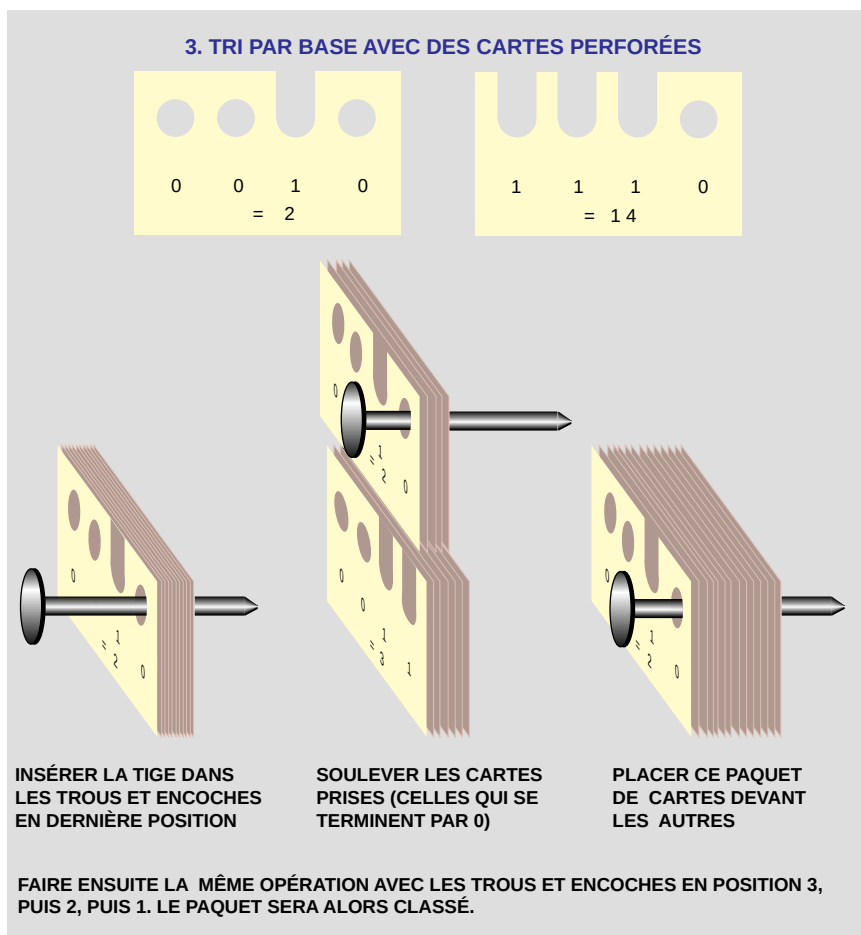
Le classement de fiches étant maintenant fait dans la mémoire des ordinateurs, ce tri est moins pratiqué. Cependant Thomas Cormen fait remarquer que, pour trier un million de nombres de 64 chiffres binaires, un tel tri (programmé et non plus bien sûr effectué avec des cartes perforées) supporte bien la comparaison en temps de calcul avec le tri-rapide. Il ne faut donc pas l'oublier.

La comparaison sur critère mathématique (comme nous l'avons faite entre le tri par insertion et le tri-rapide) n'est pas facile, car le tri par base utilise la décomposition en base 2 des nombres (l'obtenir est bien sûr un travail). De plus, on ne compare que des chiffres binaires et non plus des nombres. On ne peut donc pas évaluer le tri par base par la méthode utilisée précédemment, il faut mener des analyses plus fines. On montre cependant qu'une fois que la taille maximale des nombres à trier est fixée, le temps de calcul du tri par base est proportionnel au nombre de données à trier : il faut kn étapes de calcul pour trier n données, k étant une constante qui dépend de la taille maximale des données à trier.

En apparence, cela semble signifier qu'il faut toujours préférer cet algorithme aux deux précédents. Mais la constante de proportionnalité k étant assez grande, cette conclusion n'est vraie que pour des grandes valeurs de n .

En définitive, si vous devez réaliser un tri de données, choisir le bon algorithme n'est pas simple et demande une analyse préalable. Néanmoins la règle suivante devrait suffire dans presque toutes les situations :

– (a) s'il y a peu de données à trier (moins de 100), utilisez le tri par inser-



tion, facile à écrire et pas si mauvais que cela ;

– (b) sinon, utilisez le tri-rapide (avec mélange préalable), sauf si vous avez une très grande quantité de nombres à traiter et qu'ils restent dans un petit intervalle, auquel cas vous devez tester le tri par base.

TRI PAR SÉPARATIONS ET RETOURNEMENTS

Avec des cartes à jouer, le tri par base s'adapte très bien pour le jeu de 32 cartes. On considère qu'on veut arriver à un jeu classé comme suit : 7, 8, 9, 10, valet, dame, roi et as, en plaçant d'abord les ♥, puis les ♦, puis les ♠ et enfin les ♣. La carte numéro 0 est donc le 7♥ et la carte numéro 31 l'as ♣.

En pratique, le tri par base conduit au procédé suivant. On prend le jeu face en dessous. On retourne les cartes une par une en plaçant face au-dessus à gauche les cartes 7, 9, valet et roi (ce sont les cartes dont le numéro écrit en binaire se termine par 0) et à droite les autres. Une fois ce premier passage de toutes les cartes, on place le paquet de gauche sur celui de droite, on retourne le tout et l'on effectue un

second passage en mettant cette fois les 7, 8, valet et dame à gauche, et les autres cartes à droite. On regroupe les deux paquets comme précédemment. Pour le troisième passage, on place les 7, 8, 9 et 10 à gauche, pour le quatrième les couleurs cœur et pique à gauche et pour le cinquième et dernier passage les rouges à gauche. Le jeu est alors trié du 7♥ à l'as ♣.

LE TRI PAR SÉPARATIONS SANS RETOURNEMENT

L'adaptation directe du tri par base au jeu de 32 cartes que nous venons de décrire présente un défaut qui en ralentit la pratique : il faut retourner les cartes une à une, ce qui retarde la décision qu'on doit prendre de la placer à droite ou à gauche. À bien y réfléchir donc, un tri de cartes utile fondé sur un principe de séparations successives évitant cet inconvénient doit être possible.

Voici une telle méthode en 5 passages que j'ai trouvée par tâtonnements et dont je ne connais pas l'adaptation au jeu de 52 cartes. On prend le jeu face vers soi et on le partage cinq fois de suite en deux paquets en faisant simplement glisser les cartes, une à une, à droite ou à gauche.

– premier passage : mettre les cartes 7, 9, V et R à gauche et les autres (8 10 D A) à droite ;

– deuxième passage : mettre les cartes 9, 10, R, A à gauche et les autres (7, 8, V, D) à droite ;

– troisième passage : mettre les cartes 7, 8, 9, 10 à gauche et les autres (V, D, R, A) à droite ;

– quatrième passage : mettre les cartes ♣, ♦ à gauche et les autres ♥, ♠ à droite ;

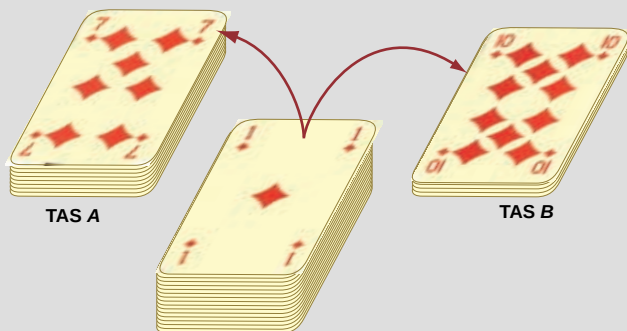
– cinquième passage : mettre les rouges à gauche et les noires à droite.

LE TRI PAR SÉPARATIONS SANS TABLE

Voici un autre perfectionnement du tri par base pour les cartes (que j'ai encore trouvé par tâtonnements) ; il permet de trier sans table les cartes d'un jeu de 32 cartes en faisant simplement passer les cartes plusieurs fois de suite d'une main à l'autre et en ne jouant que sur la possibilité de placer la carte qu'on déplace en dessous ou au-dessus du paquet. Voici les séparations à effectuer :

– premier passage : mettre les cartes 8, 9, dame et roi au-dessus et 7, 10, valet et as en dessous ;

4. TRI D'UN JEU DE 32 CARTES SANS RETOURNEMENT



PASSAGE 1 7 9 V R À GAUCHE (A)
8 10 D As À DROITE (B)
PLACER LE TAS A SUR LE TAS B

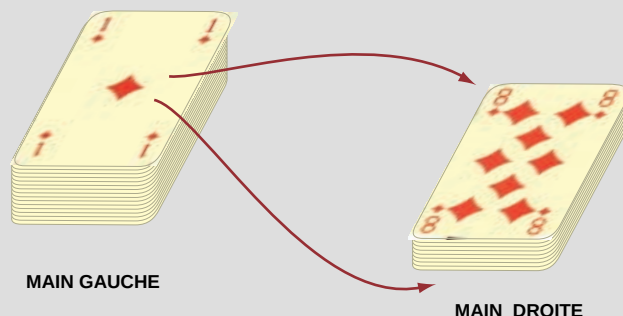
PASSAGE 2 9 10 R As À GAUCHE (A)
7 8 V D À DROITE (B)
PLACER LE TAS A SUR LE TAS B

PASSAGE 3 7 8 9 10 À GAUCHE (A)
V D R As À DROITE (B)
PLACER LE TAS A SUR LE TAS B

PASSAGE 4 ♣ ♦ À GAUCHE (A)
♥ ♠ À DROITE (B)
PLACER LE TAS A SUR LE TAS B

PASSAGE 5 ROUGES À GAUCHE (A)
NOIRES À DROITE (B)
PLACER LE TAS A SUR LE TAS B

5. TRI D'UN JEU DE 32 CARTES SANS TABLE



PASSAGE 1 8 9 D R AU-DESSUS
7 10 V As EN DESSOUS

PASSAGE 2 9 10 V D AU-DESSUS
7 8 R As EN DESSOUS

PASSAGE 3 7 8 9 10 DE ♦ ET DE ♣
V D R As DE ♥ ET DE ♠ AU-DESSUS
7 8 9 10 DE ♥ ET DE ♠
V D R As DE ♦ ET DE ♣ EN DESSOUS

PASSAGE 4 ♠ ET ♦ AU-DESSUS
♣ ET ♥ EN DESSOUS

PASSAGE 5 ROUGES AU-DESSUS
NOIRES EN DESSOUS