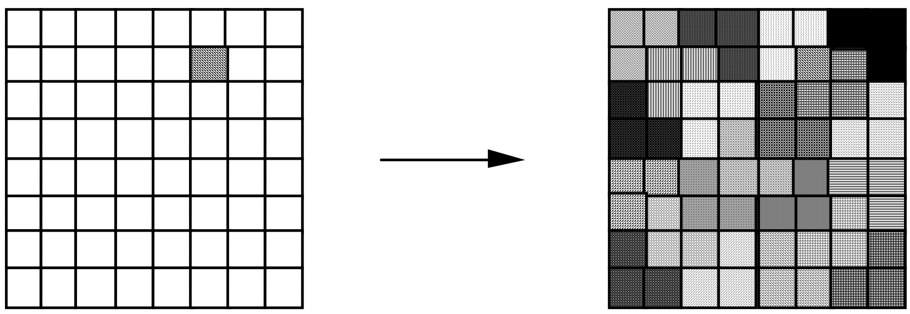
TD 1 (avec corrigé)
Diviser pour Régner

Exercice 1. Pavage particulier d'un carré

On considère un échiquier carré de côté $n = 2^m$ avec $m \geq 0$ et les quatre motifs ci-après composés d'un carré 2×2 auquel une case a été enlevée :

On se pose le problème de recouvrir **intégralement** l'échiquier, **à l'exception d'une case spéciale donnée** de coordonnées (x, y), à l'aide des motifs précédents, de sorte que **chaque case soit recouverte par un seul motif**.

Exemple. On a un échiquier de côté $8 = 2^3$ et la case spéciale est en (2, 6). Les motifs ont été distingués uniquement par souci d'identification.



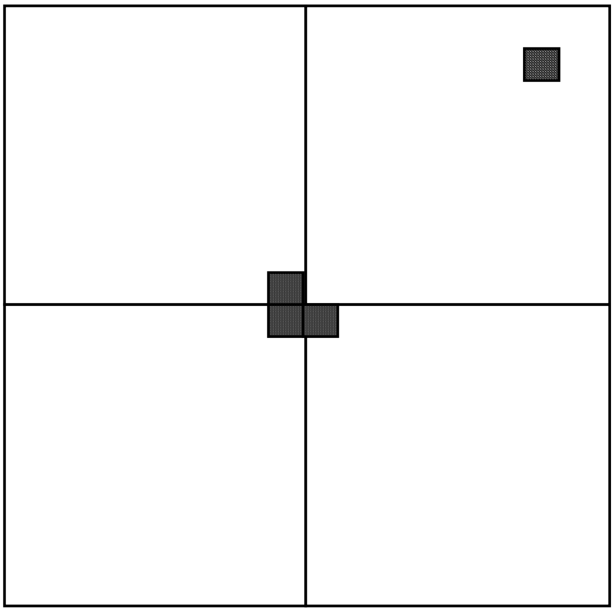
Le but de l'exercice est de concevoir une solution de type "diviser pour régner" au problème posé.

1. Proposer un modèle de division permettant de résoudre ce problème. Exprimer la division du problème initial Recouvre(2^m) **en faisant apparaître des problèmes de même nature** et **la pose de motif(s)**. On précisera les problèmes élémentaires.

Réponse.

L'idée est de diviser le carré initial en 4. L'un d'eux contient la case retirée d'origine et reste alors à faire en sorte que les 3 autres carrés de côté n/2 aient aussi une case retirée. Pour ce faire, il suffit de considérer le triplet de cases au cœur du carré initial associées aux carrés non encore amputés d'une case.

Exemple.



Recouvre(2^m) $\rightarrow$ 4 Recouvre(2^{m-1}) + pose d'un motif élémentaire

Recouvre(2^1) est élémentaire puisqu'il suffit de choisir le motif qui complète le carré de côté 2 (à 4 cases), dans lequel une case spéciale a été désignée. On peut aussi choisir 2^0 qui correspond à un carré où il n'y a que la case spéciale.

2. Établir que la **complexité temporelle exacte** de la procédure associée est égale à $(n^2 - 1)/3$ si on prend **la pose de motif comme opération élémentaire**.

Réponse.

Si Nbm(m) désigne la complexité de la procédure de recouvrement d'un échiquier de côté 2^m (c'est-à-dire le nombre de motifs utilisés pour le recouvrement), on a l'équation de récurrence :

Nbm(m) = 4 * Nbm(m – 1) + 1 et Nbm(1) = 1

Dont la solution est : Nbm(m) = $\sum_{i=0}^{m-1} 4^i = \frac{4^m - 4^0}{4-1} = \frac{4^m - 1}{3} = \frac{(2^m)^2 - 1}{3} = \frac{n^2 - 1}{3}$.

3.a. Donner une **condition nécessaire** (relative à n, le côté de l'échiquier) pour que le problème posé ait une solution (indépendamment de la façon de le résoudre).

Réponse.

Le problème a une solution seulement si (n^2 - 1), i.e. le nombre de cases à recouvrir, est un multiple de 3.

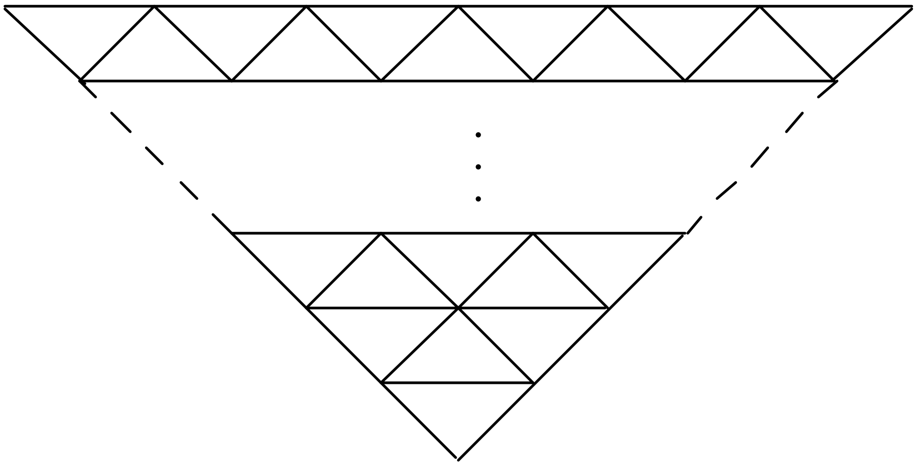
b. Quand cette condition est satisfaite, peut-on appliquer la méthode de type "diviser pour régner" proposée auparavant ?

Réponse.

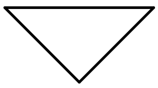
Quand cette condition est satisfaite initialement, la méthode de type "diviser pour régner" proposée auparavant ne peut être appliquée que si les sous-problèmes engendrés la satisfont aussi (ce qui est vrai pour tout côté de la forme 2^m).

Exercice 2. Dessin en "triangles"

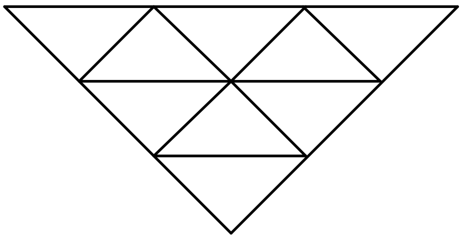
On considère le dessin ci-dessous :



qui peut être vu comme un "empilement" de lignes composées de triangles rectangles isocèles élémentaires de hauteur d ayant la forme suivante :



On appelle n le nombre "d'étages" sur lesquels des triangles sont empilés. Ainsi dans l'exemple ci-après, on a trois étages et donc $n = 3$.



1. Pour un dessin ayant n "étages", quel est le nombre de triangles élémentaires à tracer ?

Réponse.

1. Un dessin à "n étages" est composé d'une première ligne ayant un triangle élémentaire, puis d'une seconde ligne avec 2 triangles élémentaires, etc jusqu'à la dernière ligne qui comprend n triangles élémentaires. On a donc $1 + 2 + ... + n - 1 + n = n(n + 1) / 2$ triangles élémentaires à tracer.

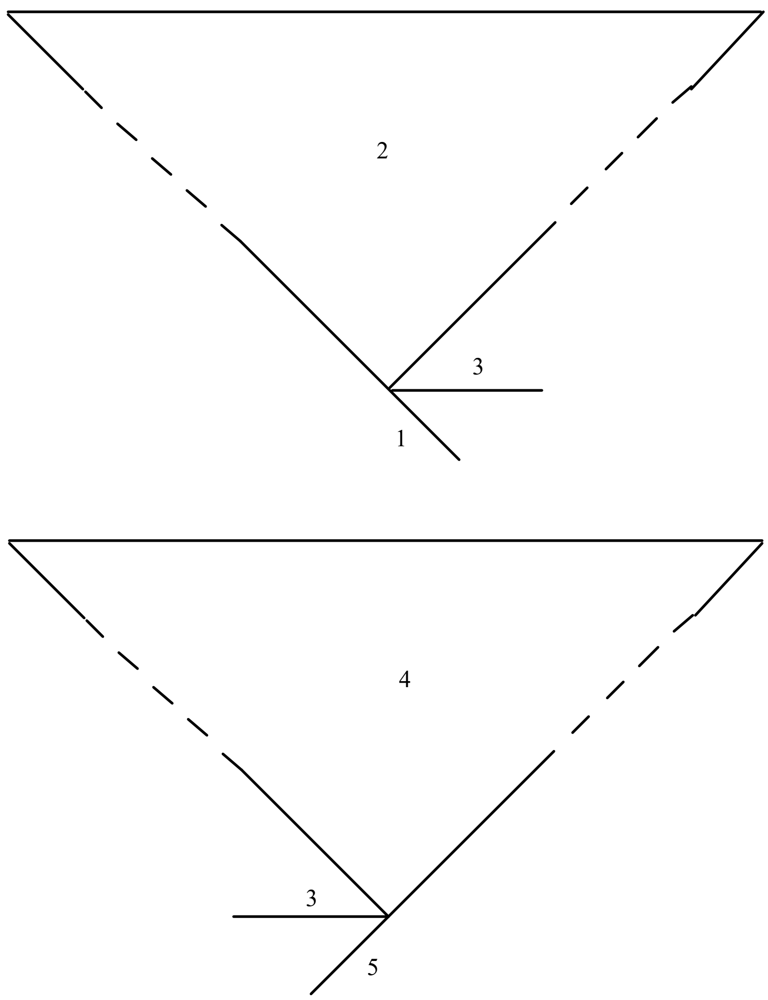
2. On envisage d'effectuer ce dessin avec un **algorithme de type diviser pour régner** dans lequel on fait appel à deux problèmes de même nature. Expliquer le principe de cette solution et en donner le modèle de division. En supposant disponible une procédure tracer(x, y) qui trace le segment allant du point courant où se trouve la plume jusqu'au point de coordonnées (x, y), écrire la procédure dessin-en-triangles correspondante d'en-tête :

proc dessin-en-triangles **c'est fixe** (**ent** n, **réel** x, y)

où n est le nombre d'étages et le couple (x, y) désigne les coordonnées de la pointe du triangle inférieur du dessin à effectuer.

Réponse.

Le principe de cette solution est illustré par le schéma ci-après :



Les numéros indiquent l'ordre des segments (1, 3 et 5) tracés ou des régions (2 et 4) dans lesquelles le dessin est effectué.

On a alors le modèle de division ci-après :

Pb(n) $\rightarrow$ 2 Pb(n – 1) + tracé d'un triangle élémentaire (en 3 temps)
Pb(1) élémentaire

Code de la procédure associée :

```
proc dessin_en_triangles c'est fixe (ent n, réel x, y)
début si n > 0 alors   tracer(x – d, y + d);
                       dessin_en_triangles(n – 1, x – d, y + d);
                       tracer(x + d, y + d);
                       dessin_en_triangles(n – 1, x + d, y + d);
                       tracer(x, y)
      fsi
fin
```

3. Combien de triangles élémentaires vont être dessinés par cette méthode ? Conclure quant à l'optimalité de cette méthode de tracé.

Réponse.

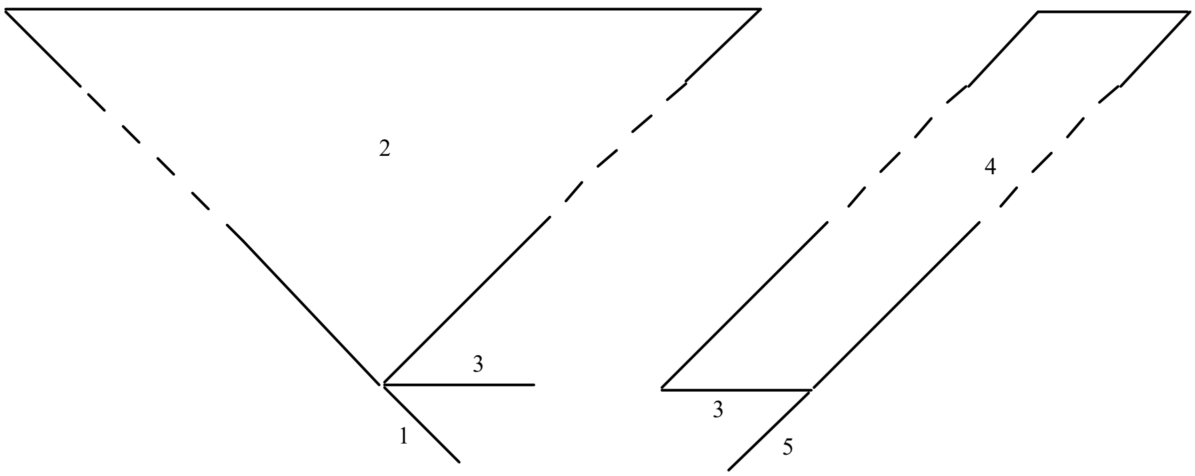
Nbtr(n) = 2 * Nbtr(n – 1) + 1
Nbtr(1) = 1

$$\begin{aligned}\text{Nbtr}(n) &= 2 * \text{Nbtr}(n-1) + 1\\ &= 2 * (2 * \text{Nbtr}(n-2) + 1) + 1\\ &= 2^{n-1} * \text{Nbtr}(1) + 2^{n-2} + 2^{n-3} + \ldots + 2 + 1\\ &= 2^{n-1} + 2^{n-2} + 2^{n-3} + \ldots + 2 + 1\\ &= 2^n - 1.\end{aligned}$$

4. Modifier la méthode de tracé précédente en **utilisant de façon conjointe deux procédés s'appuyant sur le principe diviser pour régner** (dont on explicitera le modèle de division), de sorte que la méthode obtenue soit optimale (on devra le montrer).

Réponse.

L'idée est de ne pas effectuer un appel à dessin-en-triangles la seconde fois mais à une procédure bandeau (B) qui effectue le dessin des triangles (il y en a (n – 1) en diagonale) dans la région étiquetée 4 dans le schéma ci-après.



La procédure B est elle-même fondée sur diviser pour régner, du type :

B(n) → B(n – 1).

Le modèle de division de l'autre procédure est :

Pb(n) → Pb(n – 1) + B(n – 1) + tracé d'un triangle élémentaire.

proc triangles **c'est fixe** (**ent** n, **réel** x, y)
début si n > 0 **alors** tracer(x – d, y + d);
triangles(n – 1, x – d, y + d);
tracer(x + d, y + d);
bandeau(n – 1, x, y) ;
tracer(x, y)
fsi
fin

proc bandeau **c'est fixe** (**ent** n, **réel** x, y)
début si n > 0 **alors** tracer(x – d, y + d);
tracer(x + d, y + d);
bandeau(n – 1) ;
tracer(x, y) ;
fsi
fin

Preuve de l'optimalité.

Pour n = 1 on ne trace qu'un triangle. Supposons que l'on trace n(n – 1) / 2 triangles au rang n – 1. Au rang n, on appelle Pb(n – 1) qui trace n(n – 1) / 2 triangles élémentaires et B(n – 1) qui en trace n – 1, et on trace un triangle, d'où le nombre global de :

$n(n - 1) / 2 + (n - 1) + 1 = n^2 / 2 - n / 2 + n = n^2 / 2 + n / 2 = n(n + 1) / 2$.

Plus formellement :

$T(n) = T(n-1) + B(n-1) + 3$
$T(0) = 0$

$B(n) = B(n-1) + 3$
$B(0) = 0$

On obtient :

$B(n) = 3n$ et donc :

$T(n) = T(n-1) + 3(n-1) + 3 = T(n-1) + 3n$
$T(0) = 0$

$T(n) = T(n-2) + 3n + 3(n-1) = T(n-3) + 3n + 3(n-1) + 3(n-2)$

Hypothèse : $\forall k, T(n) = T(n-k) + 3\Sigma_{i=1..k} (n-(i-1))$

En particulier pour $k = n$, on a :

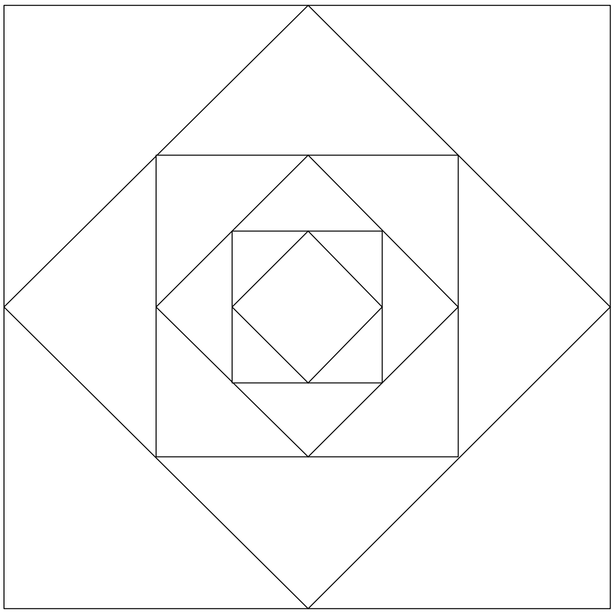
$T(n) = 3\Sigma_{i=1..n}\, n-(i-1) = 3(\Sigma_{i=1..n} (n-i)) + 3n = 3(\Sigma_{i=0..n-1}\, i) + 3n$
$= 3\Sigma_{i=0..n}\, i$
$= 3 * n(n+1)/2$

Ca se démontre facilement par récurrence.

Comme chaque appel à tracer trace le côté d'un triangle élémentaire, on trace au total $n(n+1)/2$ triangles. La procédure est optimale.

Exercice 3. Le dessin en "doubles carrés" imbriqués.

On veut définir un algorithme pour tracer le dessin ci-dessous dans lequel le motif de base est constitué de deux carrés imbriqués :



On dispose de deux primitives : *placer(x, y)* qui positionne la plume au point de coordonnées *(x, y)* et *tracer(x, y)* qui trace le segment allant du point courant au point *(x, y)*. Le dessin doit être effectué sous les contraintes suivantes : c1) à la fin du tracé la plume se trouve au point de départ, c2) la plume ne doit pas être levée, ni un trait fait plusieurs fois (pas de temps mort et pas de travail inutile).

1. Proposer une stratégie de type DpR pour réaliser ce tracé.

Ici, il n'est pas possible de dessiner le double carré extérieur (Tracé Externe), puis d'attaquer les carrés plus internes (P(n-1)), car quel que soit le point de départ du premier tracé, il ne pourra servir de départ au tracé suivant, donc on ne peut appliquer :

$$P(n) \rightarrow TE; P(n-1)$$

On va plutôt adopter le schéma :

$$P(n) \rightarrow \text{commencer } TE; P(n-1); \text{terminer } TE$$

Comme on veut employer la récursivité, il faut que le tracé de TE et de P(n – 1) s'effectuent de façon similaire, donc que l'on retrouve au cours du tracé de TE le point servant de départ d'un motif. On remarque que les points d'intersection entre un motif externe et le motif interne suivant sont au nombre de quatre : les coins du carré externe du motif interne qui sont aussi les milieux des côtés du carré interne du motif externe. Il y a donc quatre points de départ possibles (les quatre coins).

Exemple de stratégie de tracé : on part du coin haut gauche; on va vers la droite jusqu'au milieu du côté où on entame le tracé du carré intérieur; on part vers le bas gauche pour la moitié du côté et on se retrouve au point de départ pour le motif suivant; en retour de récursivité, on terminera le carré intérieur puis extérieur.

2. Donner le code de la procédure réalisant le tracé demandé.

L'algorithme suivant est conforme à la stratégie vue plus haut :

```
proc descarrés c'est fixe(ent n, réel x, y, h)
début si n > 0 alors tracer(x + h/2, y);
                     tracer(x + h/4, y – h/4);
                     descarrés(n – 1, x + h/4, y – h/4, h/2);
                     tracer(x, y – h/2);
                     tracer(x + h/2, y – h);
                     tracer(x + h, y – h/2);
                     tracer(x + h/2, y);
                     tracer(x + h, y);
                     tracer(x + h, y – h);
                     tracer(x, y – h);
                     tracer(x, y)
      fsi
fin
```

```
Appel : x0 ← ; y0 ← ; h ← ; n ← ;
        placer(x0,y0);
        descarrés(n, x0, y0, h);
```

3. Donner la complexité de cette procédure en termes d'appels à *tracer*.

$D(n) = D(n – 1) + 10$
$D(1) = 10$

donc $D(n) = 10 * n$ (complexité linéaire)

4. Déterminer la longueur du tracé effectué en fonction de *n* et de *l*.

$lgtracé(n, h) = 4l + 2h\sqrt{2} + lgtracé(n-1, h/2)$
$lgtracé(0, l) = 0$

Donc $lgtracé(n, h) = (4h + 2h\sqrt{2}) (1 + 1/2 + 1/4 + ... + 1/(2^{n-1}))$
$= (4h + 2h\sqrt{2})(((1 – (1/2)^n)/(1 – 1/2))$
$= 4h (2 + \sqrt{2}) (1 – (1/2)^n)$

Quand n tend vers l'infini lgtracé(n, h) tend vers $4h(2+\sqrt{2})$ (qui est en $\theta(h)$).

Exercice 4. La bâtière.

On considère un tableau "carré" de côté $c = 2^p$ de valeurs entières positives telles que les valeurs d'une même ligne et celles d'une même colonne sont **ordonnées de façon strictement croissante**. Un exemple d'un tel tableau est donné ci-dessous pour c = 4 (p = 2) :

2	4	25	28
3	10	28	30
7	15	32	43
20	28	36	57

On étudie le problème lié à la recherche d'une valeur v fixée dans un tel tableau B. À cette fin, on envisage une solution de type "diviser pour régner" en distinguant les valeurs :

u1 = B[c **div** 2, c **div** 2] et u2 = B[(c **div** 2) + 1, (c **div** 2) + 1]

div représentant l'opérateur de division entière. Dans l'exemple ci-dessus, u1 = 10 et u2 = 32.

1. Montrer que si la valeur recherchée v est strictement inférieure à u2, on peut éliminer une partie du tableau B (à préciser) pour poursuivre la recherche. Préciser de même, ce qu'il est possible de faire quand la valeur v est strictement supérieure u1.

Réponse.

Si v > u1, on peut éliminer le quart supérieur gauche du tableau B puisque par définition toute valeur de cette partie est inférieure ou égale à u. Si v < u2, on peut par analogie supprimer le quart inférieur droit de B.

2. En déduire un modèle de résolution de type "diviser pour régner" en réduction logarithmique de la forme :

$$Pb(n) \rightarrow a\ Pb(n/b) + f(n)$$

en précisant le(s) problème(s) élémentaire(s), les valeurs de a et b, ce que représente n et ce que réalise la "fonction de collage" f en particulier en ce qui concerne la localisation de v en fonction des localisations retournées par les sous-problèmes engendrés.

Réponse.

Le rôle de f est notamment de tester v par rapport à u1 et u2. En cas d'égalité, il y a arrêt de la récursivité (avec localisation appropriée), sinon on poursuit avec les trois sous-tableaux de dimension n/2 convenables.

Compte tenu de ce qui a été dit en 1, le modèle de division est ici :

$Pb(c) \rightarrow 3\ Pb(c/2) + test(u1, u2, v) \quad c > 1$

Pb(1) élémentaire.

En retour de récursivité, on a à distinguer 3 cas :

a) 3 échecs, i.e., couples (-1, -1) et localisation rendue = (-1, -1)

b) 1 seul succès dont la valeur de localisation est retournée

c) au - 2 succès et la valeur de localisation retournée est l'une d'elles (peu importe).

3. Établir la classe de complexité au pire de cette méthode (en terme de nombre de comparaisons).

Réponse.

On a :

$T(n) = 3\ T(n/2)$ + au plus 2 comparaisons
$T(1) = 1$

ce qui correspond à une situation vue en cours où la complexité de f est constante (cas 1 avec a > b) et donc la complexité au pire de la procédure "recherche" est en $\theta(n^{\log_2 3}) \approx \theta(n^{1.59})$.

4. Donner le principe permettant d'adapter cette stratégie au cas d'un tableau B ayant q lignes et r colonnes avec q et r quelconques.

Réponse.

On complète le tableau considéré avec une valeur conventionnelle (∞) pour arriver à un tableau carré de côté c tel que c est la plus petite valeur vérifiant : $c = 2^p$, $c \geq q$, $c \geq r$.