

Fiche formules + règles - Mathématiques BTS SIO

Graphes, matrices, numérisation et algorithmique

partiel	formules	règles	méthodes	exemples
---------	----------	--------	----------	----------

Source : Synthèse construite à partir des 9 JPEGS de cours, du fichier_Math.md et des chapitres pertinents du PDF Ellipses : arithmétique p.41-46, matrices p.73-79, graphes p.85-97. Le PDF ne couvre pas l'algorithmique appliquée ; cette partie est donc complétée à partir des photos et du programme visé.

0. Carte mentale du partiel

Algorithmique	Écrire et lire un pseudo-code ; suivre les variables ; boucles Pour et Tant que ; Fibonacci.
Graphes	Lire un graphe orienté ; arcs ; chemins ; boucles ; circuits ; accessibilité.
Matrices	Construire une matrice d'adjacence ; interpréter M^2, M^3, M^4 ; somme
Matrices générales	$M + M^2 + M^3 + M^4$.
Numérisation	Addition, produit ligne-colonne, identité I , inverse, résolution $AX = Y$.
Numérisation	Bases 2, 10, 16 ; conversions ; addition binaire.

Règle de travail rapide : pour chaque question, écrire d'abord la méthode, puis faire le calcul. **Beau-coup de points viennent d'une procédure correcte même si le calcul final contient une petite erreur.**

1. Algorithmique - comprendre, lire, écrire

Source : Cette partie comble le manque du PDF : l'ouvrage indique qu'il ne traite pas le module d'algorithmique appliquée. Elle vient donc des JPEGS et du fichier_Math.md.

1.1 Vocabulaire de base

Variable	Zone mémoire nommée : a, b, c, i . Elle contient une valeur qui peut changer.
Déclaration	On annonce le type des variables : i, a, b, c : Entier.
Affectation	$a \leftarrow 1$ signifie : on met la valeur 1 dans la variable a . Ce n'est pas une équation.
Calcul	$c \leftarrow a + b$ signifie : calculer $a + b$, puis stocker le résultat dans c .
Affichage	Afficher(c) écrit la valeur actuelle de c .
Compteur	Variable, souvent i , qui sert à compter le nombre de tours d'une boucle.
Trace	Tableau qui montre les valeurs successives des variables après chaque instruction ou chaque tour.

Différence essentielle : $a = 1$ est une égalité mathématique ; $a \leftarrow 1$ est une instruction qui modifie le contenu de la variable a .

1.2 Lire un algorithme : méthode en 5 étapes

- Repérer les variables déclarées et leur type : entiers, réels, booléens, etc.
- Repérer les valeurs initiales : par exemple $a \leftarrow 1, b \leftarrow 1, i \leftarrow 1$.
- Repérer le type de boucle : Pour ou Tant que.
- Faire un tableau de trace : une ligne par tour de boucle.
- Noter les valeurs affichées exactement au moment où l'instruction Afficher apparaît.

1.3 Boucle Pour

Règle : utiliser une boucle Pour quand le nombre de répétitions est connu avant de commencer.

```

Pour i allant de 1 à 8 faire
    instructions
FinPour
```

Cela signifie que les instructions sont exécutées pour :

$$i = 1, 2, 3, 4, 5, 6, 7, 8.$$

Il y a donc 8 tours.

Déclencheur examen : "Compléter le tableau d'exécution" ou "Que va afficher l'algorithme ?"

Méthode : écrire les colonnes i, a, b, c , puis remplir ligne par ligne.

1.4 Boucle Tant que

Règle : utiliser une boucle Tant que quand la répétition dépend d'une condition.

```

i <- 1
Tant que i <= 8 faire
    instructions
    i <- i + 1
FinTantQue
```

Erreur classique : oublier $i \leftarrow i + 1$. Si i ne change pas, la condition peut rester vraie et la boucle peut ne jamais s'arrêter.

1.5 Fibonacci : idée mathématique

La suite de Fibonacci commence par deux valeurs initiales, puis chaque nouveau terme est la somme des deux précédents :

$$F_1 = 1, \quad F_2 = 1, \quad F_{n+2} = F_{n+1} + F_n.$$

Donc :

$$1, 1, 2, 3, 5, 8, 13, 21, 34, \dots$$

1.6 Fibonacci : pseudo-code avec boucle Pour

```

Algorithme Fibonacci
Déclaration
    i, a, b, c : Entier
Début
    a <- 1
    b <- 1
    Afficher(a)
    Afficher(b)
    Pour i allant de 1 à 8 faire
        c <- a + b
        Afficher(c)
        b <- a
        a <- c
    FinPour
Fin
```

Pourquoi l'ordre est important : on calcule d'abord $c \leftarrow a + b$. Ensuite seulement, on décale les anciennes valeurs : $b \leftarrow a$, puis $a \leftarrow c$.

1.7 Trace Fibonacci

Si au début $a = 1$ et $b = 1$:

tour	a avant	b avant	$c = a + b$	valeur affichée
1	1	1	2	2
2	2	1	3	3
3	3	2	5	5
4	5	3	8	8
5	8	5	13	13

1.8 Guide pour écrire un algorithme propre

- Donner un nom : Algorithme Nom.
- Déclarer toutes les variables : i, a, b, c : Entier.
- Initialiser les variables avant la boucle.
- Choisir la bonne boucle : Pour si le nombre de tours est connu ; Tant que si on teste une condition.
- Dans une boucle Tant que, modifier la variable de condition.
- Afficher uniquement ce qui est demandé.
- Vérifier avec une petite trace de 2 ou 3 tours.

2. Graphes orientés

Source : PDF Ellipses : Chapitre 5, graphes orientés, chemins, boucles, circuits, matrice d'adjacence et fermeture transitive.

2.1 Vocabulaire

Sommet	Point du graphe : A, B, C, D .
Arc	Flèche orientée : $A \rightarrow B$.
Chemin	Suite de sommets reliés par des arcs dans le bon sens.
Boucle	Arc qui part d'un sommet et revient au même sommet. Longueur 1.
Circuit	Chemin fermé : le sommet de départ est aussi le sommet d'arrivée.
Longueur	Nombre d'arcs utilisés dans le chemin.

Règle à retenir : dans un graphe orienté, $A \rightarrow B$ ne signifie pas automatiquement $B \rightarrow A$. Il faut vérifier le sens de chaque flèche.

2.2 Vérifier un chemin

Un chemin

$$(S_0, S_1, S_2, \dots, S_n)$$

existe si tous les arcs suivants existent :

$$S_0 \rightarrow S_1, \quad S_1 \rightarrow S_2, \quad \dots, \quad S_{n-1} \rightarrow S_n.$$

Exemples :

- (A, B, A) existe si $A \rightarrow B$ et $B \rightarrow A$ existent.
- (A, C, D) existe si $A \rightarrow C$ et $C \rightarrow D$ existent.
- (B, A, B) est un circuit de longueur 2 si $B \rightarrow A$ et $A \rightarrow B$ existent.

3. Matrice d'adjacence

Source : PDF Ellipses : Chapitre 5, matrice d'adjacence et puissances de matrices.

3.1 Définition

Pour un graphe orienté de sommets $S_1, S_2, \dots, S_n$, la matrice d'adjacence est :

$$M = (m_{ij})_{1 \leq i, j \leq n}$$

avec

$$m_{ij} = \begin{cases} 1 & \text{s'il existe un arc } S_i \rightarrow S_j, \\ 0 & \text{sinon.} \end{cases}$$

Règle à apprendre : ligne = sommet de départ ; colonne = sommet d'arrivée.

3.2 Exemple type

Avec l'ordre des sommets A, B, C, D :

$$M = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

Interprétation :

- $m_{AB} = 1$: il existe l'arc $A \rightarrow B$.
- $m_{AC} = 1$: il existe l'arc $A \rightarrow C$.
- $m_{AD} = 0$: il n'existe pas d'arc $A \rightarrow D$.
- $m_{DA} = 1$: il existe l'arc $D \rightarrow A$.

3.3 Méthode de construction

- Fixer l'ordre des sommets : A, B, C, D .
- Tracer mentalement les lignes : ligne A , ligne B , ligne C , ligne D .
- Pour chaque ligne, demander : depuis ce sommet, vers quels sommets partent les flèches ?
- Mettre 1 si la flèche existe, 0 sinon.

4. Puissances de matrices et chemins

Source : PDF Ellipses : Chapitre 5, matrice d'adjacence et puissance de matrice ; fermeture transitive.

Règle fondamentale : dans une matrice d'adjacence M , le coefficient (i, j) de M^k donne le nombre de chemins de longueur exactement k allant de S_i vers S_j .

$$(M^k)_{ij} = \text{nombre de chemins de longueur } k \text{ de } S_i \text{ vers } S_j.$$

M	Chemins de longueur 1 : arcs directs.
M^2	Chemins de longueur 2.
M^3	Chemins de longueur 3.
M^4	Chemins de longueur 4.

4.1 Accessibilité

Pour savoir si un sommet peut atteindre un autre en plusieurs étapes, on peut regarder une somme de puissances :

$$M + M^2 + M^3 + M^4.$$

Si le coefficient (i, j) de cette somme est non nul, alors S_j est accessible depuis S_i avec une longueur comprise entre 1 et 4.

4.2 Fermeture transitive

La fermeture transitive indique simplement si un sommet peut en atteindre un autre, sans forcément compter le nombre de chemins. On remplace les coefficients non nuls par 1 :

$$\text{non nul} \Rightarrow 1, \quad 0 \Rightarrow 0.$$

5. Matrices générales

Source : PDF Ellipses : Chapitre 4, opérations sur les matrices, multiplication, inverse et applications.

5.1 Formats

Une matrice de taille $m \times n$ a m lignes et n colonnes.

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}.$$

5.2 Addition

L'addition est possible seulement si les matrices ont le même format :

$$A + B = (a_{ij} + b_{ij}).$$

5.3 Multiplication par une constante

$$\lambda A = (\lambda a_{ij}).$$

Chaque coefficient est multiplié par λ .

5.4 Multiplication de matrices

Le produit

$$A_{m \times n} \times B_{n \times p}$$

est possible si le nombre de colonnes de A est égal au nombre de lignes de B . Le résultat est de taille :

$$AB \in \mathcal{M}_{m \times p}.$$

Formule :

$$(AB)_{ij} = \sum_{k=1}^n a_{ik}b_{kj}.$$

Règle pratique : coefficient (i, j) du produit = ligne i de la première matrice $\times$ colonne j de la deuxième matrice.

Exemple :

$$\begin{pmatrix} a & b & c \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = ax + by + cz.$$

5.5 Exemple numérique

$$A = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 3 & 4 \\ 5 & 6 \end{pmatrix}.$$

$$\text{Alors } AB = \begin{pmatrix} 1 \cdot 3 + 2 \cdot 5 & 1 \cdot 4 + 2 \cdot 6 \\ 0 \cdot 3 + 1 \cdot 5 & 0 \cdot 4 + 1 \cdot 6 \end{pmatrix} = \begin{pmatrix} 13 & 16 \\ 5 & 6 \end{pmatrix}.$$

Erreur fréquente : en général $AB \neq BA$. La multiplication de matrices n'est pas commutative.

6. Matrice identité et inverse

Source : PDF Ellipses : Chapitre 4, matrice inverse ; l'identité est définie comme la matrice composée de 0, sauf la diagonale principale qui contient des 1.

6.1 Matrice identité

$$I_n = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}.$$

Exemple :

$$I_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Propriété :

$$AI = IA = A.$$

6.2 Matrice inverse

Une matrice C est l'inverse de A si :

$$CA = AC = I.$$

Dans les exercices de BTS, il est souvent demandé de vérifier qu'une matrice est inverse d'une autre, en calculant un produit et en obtenant I .

6.3 Résolution type $AX = Y$

Si

$$AX = Y$$

et si on sait que

$$CA = I,$$

alors on multiplie à gauche par C :

$$CAX = CY.$$

Comme $CA = I$:

$$IX = CY.$$

Donc :

$$X = CY.$$

Raccourci examen : si $AX = Y$ et $CA = I$, alors $X = CY$. Attention : on multiplie à gauche, pas à droite.

7. Numérisation - bases 2, 10, 16

Source : PDF Ellipses : Chapitre 3, numération et changements de base ; bases 10, 2 et 16 ; conversions ; opérations.

7.1 Décomposition dans une base

Un nombre en base b s'écrit comme une somme de puissances de b :

$$(a_n a_{n-1} \dots a_1 a_0)_b = \sum_{k=0}^n a_k b^k.$$

Base 2 chiffres autorisés : 0, 1. Puissances : 1, 2, 4, 8, 16, 32, 64, 128, ...

Base 10 chiffres autorisés : 0, 1, ..., 9. **Base 16** chiffres autorisés : 0, 1, ..., 9, A, B, C, D, E, F.

$$A = 10, \quad B = 11, \quad C = 12, \quad D = 13, \quad E = 14, \quad F = 15.$$

7.2 Binaire vers décimal

Exemple :

$$11011_2 = 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0.$$

Donc :

$$11011_2 = 16 + 8 + 0 + 2 + 1 = 27_{10}.$$

7.3 Décimal vers binaire

Méthode : divisions euclidiennes successives par 2, puis lecture des restes de bas en haut.

Exemple utile :

$$185_{10} = 10111001_2.$$

Vérification :

$$10111001_2 = 128 + 32 + 16 + 8 + 1 = 185.$$

7.4 Hexadécimal vers décimal

Exemple :

$$1CE_{16} = 1 \cdot 16^2 + C \cdot 16^1 + E \cdot 16^0.$$

Comme $C = 12$ et $E = 14$:

$$1CE_{16} = 1 \cdot 256 + 12 \cdot 16 + 14 = 256 + 192 + 14 = 462_{10}.$$

7.5 Décimal vers hexadécimal

Méthode : divisions euclidiennes successives par 16, puis lecture des restes de bas en haut. Si un reste est supérieur à 9, le remplacer par la lettre correspondante.

Exemple de règle :

$$11 \mapsto B, \quad 14 \mapsto E.$$

7.6 Binaire vers hexadécimal

Méthode : regrouper les bits par paquets de 4, en partant de la droite.

$$10111001_2 = 1011 \ 1001_2 = B9_{16}.$$

8. Addition binaire

Source : PDF Ellipses : Chapitre 3, opérations en base.

8.1 Règles d'addition

$$0 + 0 = 0, \quad 0 + 1 = 1, \quad 1 + 0 = 1, \quad 1 + 1 = 10_2.$$

Avec une retenue :

$$1 + 1 + 1 = 11_2.$$

8.2 Exemple du tableau

$$11011_2 + 00101_2.$$

En décimal :

$$11011_2 = 27_{10}, \quad 00101_2 = 5_{10}.$$

Donc :

$$27 + 5 = 32.$$

Or :

$$32_{10} = 10000_2.$$

Ainsi :

$$11011_2 + 101_2 = 10000_2.$$

9. Déclencheurs de questions

"Que signifie $m_{ij} = 1$?" Il existe un arc du sommet ligne i vers le sommet colonne j .

"Construire la matrice d'adjacence" Fixer l'ordre des sommets ; ligne = départ ; colonne = arrivée ; mettre 1 ou 0.

"Interpréter M^3 " Coefficients = nombre de chemins de longueur exactement 3.

"Sommet accessible ?" Regarder $M + M^2 + M^3 + M^4$ ou la fermeture transitive.

"Calculer AB " Vérifier les formats ; faire ligne par colonne.

" $AX = Y$ et $CA = I$ " Multiplier à gauche par C ; conclure $X = CY$.

"Convertir en base 2" Divisions par 2 ou décomposition en puissances de 2.

"Convertir en base 16" Divisions par 16 ou paquets de 4 bits.

"Boucle Pour" Nombre de tours connu ; tracer les variables à chaque tour.

"Boucle Tant que" Condition ; vérifier que la variable de contrôle évolue.

10. Memory dump de début d'épreuve

A écrire sur brouillon dès le début si autorisé :

Adjacence : ligne = départ, colonne = arrivée

$(M^k)_{ij}$ = nombre de chemins de longueur k de i vers j

$$A_{m \times n} B_{n \times p} \Rightarrow AB_{m \times p}$$

$$AX = Y, \quad CA = I \Rightarrow X = CY$$

$$F_{n+2} = F_{n+1} + F_n$$

$$1 + 1 = 10_2, \quad A = 10, \quad B = 11, \quad C = 12, \quad D = 13, \quad E = 14, \quad F = 15$$