

Projet d'algorithmique

Transformée de Burrows Wheeler
Compression d'Huffman

Transformée de Burrow-Weeler (BWT)

- Publiée par Michael Burrows et David Wheeler en 1994
- Utilisée avec de grands jeux de séquences
- Transforme l'ordre des caractères d'un texte
- Sortie d'un BWT est facilement compressable
 - Utilisé par bzip2

Algorithme naïf de construction du BWT



- Ajout d'un caractère extérieur (\$) en fin de séquence S
- Générer une matrice carrée ($n \times n$) de tous les décalages de la séquence S
- Trier les lignes de la matrice par ordre lexicographique
- BWT(S) est la dernière colonne de la matrice obtenue

Exemple de construction de BWT

- Sequence : ACTTGATC
- A C T T G A T C \$

Exemple de construction de BWT

- Sequence : ACTTGATC
- A C T T G A T C \$
- \$ A C T T G A T C

Exemple de construction de BWT

- Sequence : ACTTGATC
- A C T T G A T C \$
- \$ A C T T G A T C
- C \$ A C T T G A T

Exemple de construction de BWT

- Sequence : ACTTGATC
- A C T T G A T C \$
- \$ A C T T G A T C
- C \$ A C T T G A T
- T C \$ A C T T G A

Exemple de construction de BWT

- Sequence : ACTTGATC
- A C T T G A T C \$
- \$ A C T T G A T C
- C \$ A C T T G A T
- T C \$ A C T T G A
- A T C \$ A C T T G

Exemple de construction de BWT

- Sequence : ACTTGATC
- A C T T G A T C \$
- \$ A C T T G A T C
- C \$ A C T T G A T
- T C \$ A C T T G A
- A T C \$ A C T T G
- G A T C \$ A C T T

Exemple de construction de BWT

- Sequence : ACTTGATC
- A C T T G A T C \$
- \$ A C T T G A T C
- C \$ A C T T G A T
- T C \$ A C T T G A
- A T C \$ A C T T G
- G A T C \$ A C T T
- T G A T C \$ A C T

Exemple de construction de BWT

- Sequence : ACTTGATC
- A C T T G A T C \$
- \$ A C T T G A T C
- C \$ A C T T G A T
- T C \$ A C T T G A
- A T C \$ A C T T G
- G A T C \$ A C T T
- T G A T C \$ A C T
- T T G A T C \$ A C

Exemple de construction de BWT

- Sequence : ACTTGATC



- A C T T G A T C \$
- \$ A C T T G A T C
- C \$ A C T T G A T
- T C \$ A C T T G A
- A T C \$ A C T T G
- G A T C \$ A C T T
- T G A T C \$ A C T
- T T G A T C \$ A C
- C T T G A T C \$ A

Exemple de construction de BWT

- Sequence : ACTTGATC

- A C T T G A T C \$

- \$ A C T T G A T C

- C \$ A C T T G A T

- T C \$ A C T T G A

- A T C \$ A C T T G

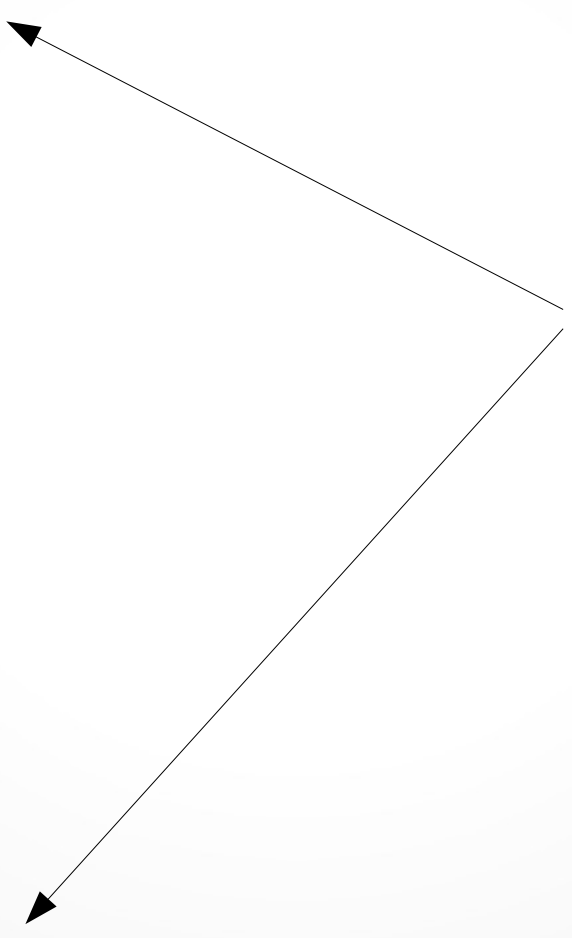
- G A T C \$ A C T T

- T G A T C \$ A C T

- T T G A T C \$ A C

- C T T G A T C \$ A

Tri des sequences par ordre
alphabetique



Exemple de construction de BWT

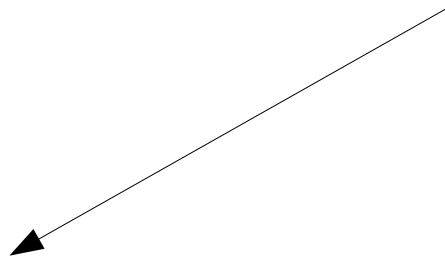
- Sequence : ACTTGATC
- \$ A C T T G A T C
- A C T T G A T C \$
- A T C \$ A C T T G
- C \$ A C T T G A T
- C T T G A T C \$ A
- G A T C \$ A C T T
- T C \$ A C T T G A
- T G A T C \$ A C T
- T T G A T C \$ A C

Exemple de construction de BWT

- Sequence : ACTTGATC

- \$ A C T T G A T **C**
- A C T T G A T C \$
- A T C \$ A C T T **G**
- C \$ A C T T G A **T**
- C T T G A T C \$ **A**
- G A T C \$ A C T **T**
- T C \$ A C T T G **A**
- T G A T C \$ A C **T**
- T T G A T C \$ A **C**

Recuperation du dernier
caractere de chaque ligne



Exemple de construction de BWT

- Sequence : ACTTGATC
-
- BWT = C\$GTATATC

Algorithme naïf de reconstruction du BWT

- Générer une matrice carrée $n \times n$ tel que
 - Insérer la séquence mélangée dans la première colonne
 - Trier les lignes par ordre lexicographique
- La séquence triée est la ligne de la matrice qui contient le symbole (\$) en fin de séquence.

Exemple de reconstruction de BWT

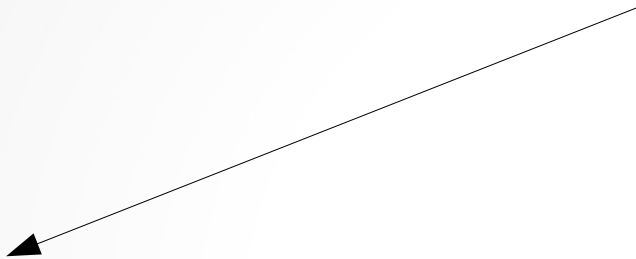
- BWT : C\$GTATATC

Exemple de reconstruction de BWT

- BWT : C\$GTATATC

- C
- \$
- G
- T
- A
- T
- A
- T
- C

Ajout du mot BWT dans la premiere colonne



Exemple de reconstruction de BWT

- BWT : C\$GTATATC

- \$
- A
- A
- C
- C
- G
- T
- T
- T



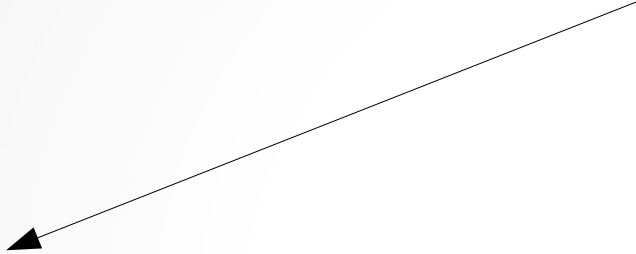
Tri des lignes par ordre lexicographique

Exemple de reconstruction de BWT

- BWT : C\$GTATATC

- C \$
- \$ A
- G A
- T C
- A C
- T G
- A T
- T T
- C T

Ajout du mot BWT dans la premiere colonne



Exemple de reconstruction de BWT

- BWT : C\$GTATATC
- \$ A
- A C
- A T
- C \$
- C T
- G A
- T C
- T G
- T T

Exemple de reconstruction de BWT

- BWT : C\$GTATATC

- C \$ A

- \$ A C

- G A T

- T C \$

- A C T

- T G A

- A T C

- T T G

- C T T

Exemple de reconstruction de BWT

- BWT : C\$GTATATC
- \$ A C
- A C T
- A T C
- C \$ A
- C T T
- G A T
- T C \$
- T G A
- T T G

Exemple de reconstruction de BWT

- BWT : C\$GTATATC

- C \$ A C

- \$ A C T

- G A T C

- T C \$ A

- A C T T

- T G A T

- A T C \$

- T T G A

- C T T G

Exemple de reconstruction de BWT

- BWT : C\$GTATATC

- \$ A C T

- A C T T

- A T C \$

- C \$ A C

- C T T G

- G A T C

- T C \$ A

- T G A T

- T T G A

Exemple de reconstruction de BWT

- BWT : C\$GTATATC

- C \$ A C T

- \$ A C T T

- G A T C \$

- T C \$ A C

- A C T T G

- T G A T C

- A T C \$ A

- T T G A T

- C T T G A

Exemple de reconstruction de BWT

- BWT : C\$GTATATC
- \$ A C T T
- A C T T G
- A T C \$ A
- C \$ A C T
- C T T G A
- G A T C \$
- T C \$ A C
- T G A T C
- T T G A T

Exemple de reconstruction de BWT

- BWT : C\$GTATATC
- C \$ A C T T
- \$ A C T T G
- G A T C \$ A
- T C \$ A C T
- A C T T G A
- T G A T C \$
- A T C \$ A C
- T T G A T C
- C T T G A T

Exemple de reconstruction de BWT

- BWT : C\$GTATATC
- \$ A C T T G
- A C T T G A
- A T C \$ A C
- C \$ A C T T
- C T T G A T
- G A T C \$ A
- T C \$ A C T
- T G A T C \$
- T T G A T C

Exemple de reconstruction de BWT

- BWT : C\$GTATATC

- C \$ A C T T G

- \$ A C T T G A

- G A T C \$ A C

- T C \$ A C T T

- A C T T G A T

- T G A T C \$ A

- A T C \$ A C T

- T T G A T C \$

- C T T G A T C

Exemple de reconstruction de BWT

- BWT : C\$GTATATC
- \$ A C T T G A
- A C T T G A T
- A T C \$ A C T
- C \$ A C T T G
- C T T G A T C
- G A T C \$ A C
- T C \$ A C T T
- T G A T C \$ A
- T T G A T C \$

Exemple de reconstruction de BWT

- BWT : C\$GTATATC

- C \$ A C T T G A

- \$ A C T T G A T

- G A T C \$ A C T

- T C \$ A C T T G

- A C T T G A T C

- T G A T C \$ A C

- A T C \$ A C T T

- T T G A T C \$ A

- C T T G A T C \$



Exemple de reconstruction de BWT

- BWT : C\$GTATATC
- \$ A C T T G A T
- A C T T G A T C
- A T C \$ A C T T
- C \$ A C T T G A
- C T T G A T C \$
- G A T C \$ A C T
- T C \$ A C T T G
- T G A T C \$ A C
- T T G A T C \$ A

Exemple de reconstruction de BWT

- BWT : C\$GTATATC
- C \$ A C T T G A T
- \$ A C T T G A T C
- G A T C \$ A C T T
- T C \$ A C T T G A
- A C T T G A T C \$
- T G A T C \$ A C T
- A T C \$ A C T T G
- T T G A T C \$ A C
- C T T G A T C \$ A

Exemple de reconstruction de BWT

- BWT : C\$GTATATC

- C \$ A C T T G A T

- \$ A C T T G A T C

- G A T C \$ A C T T

- T C \$ A C T T G A

- A C T T G A T C \$

- T G A T C \$ A C T

- A T C \$ A C T T G

- T T G A T C \$ A C

- C T T G A T C \$ A

Selection de la ligne qui contient le '\$'
comme dernier caractere



Exemple de reconstruction de BWT

- BWT : C\$GTATATC
- C \$ A C T T G A T
- \$ A C T T G A T C
- G A T C \$ A C T T
- T C \$ A C T T G A
- **A C T T G A T C \$**
- T G A T C \$ A C T
- A T C \$ A C T T G
- T T G A T C \$ A C
- C T T G A T C \$ A

Exemple de reconstruction de BWT

- BWT : C\$GTATATC
- Sequence reconstruite : A C T T G A T C

Compression d'Huffman

- Compression d'Huffman est utilisée dans de nombreux standards de compression
 - Bzip2
 - JPEG
-
- Au lieu de stocker chaque caractère comme une valeur ASCII de 8/16 ou 64 bits (système d'exploitation), l'algorithme stocke les caractères fréquents avec moins de 8/16 ou 64 bits

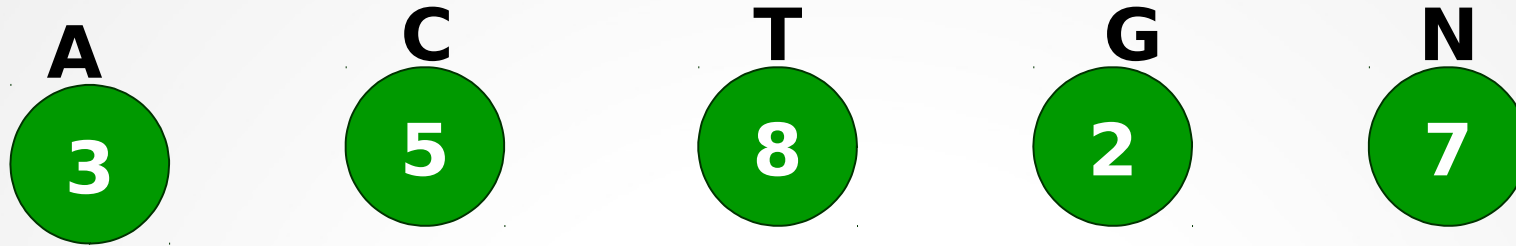
Algorithme de compression d'Huffman

- Calcul de la fréquence de chaque caractère
- Création d'un arbre binaire représentant la meilleur compression
- Compression du texte en fonction de l'arbre binaire
- Pour chaque caractère lu, lire dans l'arbre le chemin accédant à ce caractère depuis la racine
- Taille de l'encodage binaire = taille du chemin
- Sauver l'arbre binaire (pour la décompression)

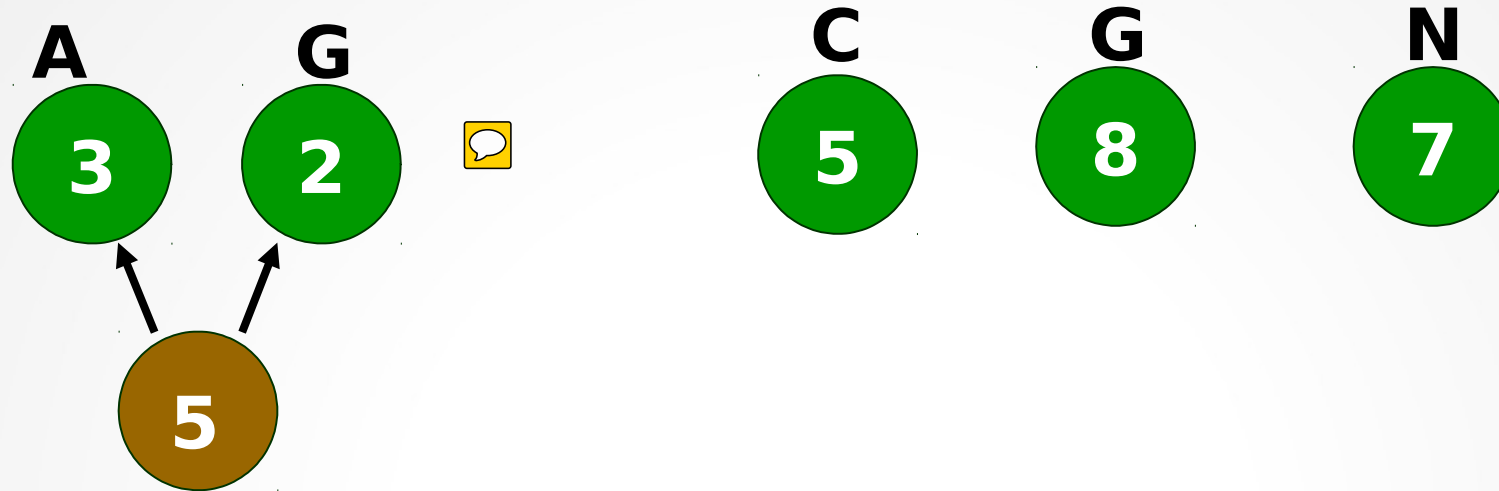
Algo de création de l'arbre binaire d'Huffman

- Initialisation
 - Chaque caractère est dans une feuille (nœud)
 - Poids de la feuille = fréquence du caractère
- Tant que tous les nœuds ne sont pas reliés
 - Sélectionne deux nœuds G et D (feuille = arbre)
 - Tel que G, D ont les deux plus petits poids
 - Créer un nouveau nœud
 - Fils gauche \rightarrow G
 - Fils droite \rightarrow D
 - Poids du nouveau nœud = $\text{Poids}(L) + \text{Poids}(R)$

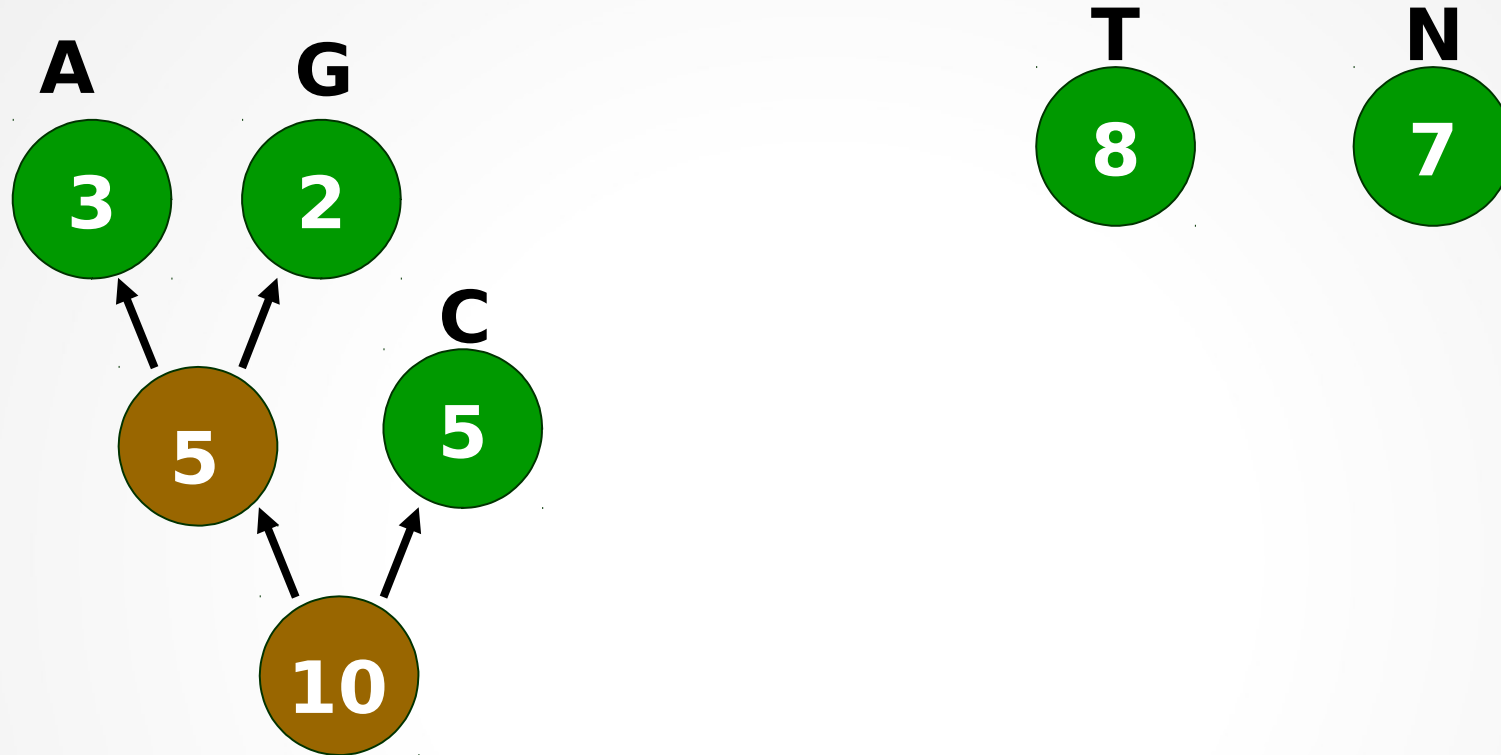
Exemple de création de l'arbre



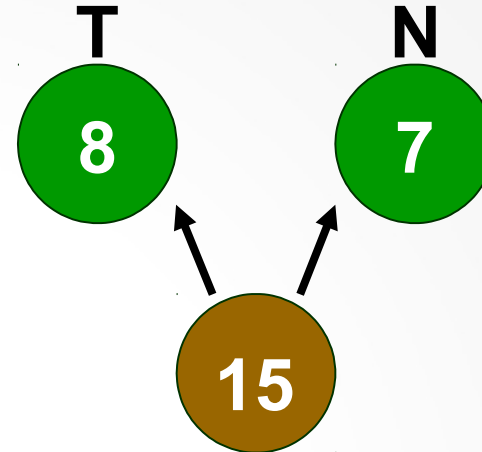
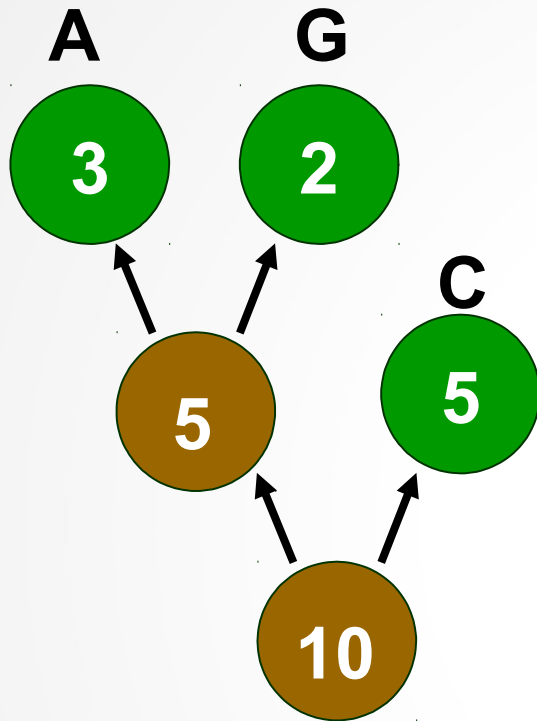
Exemple de création de l'arbre



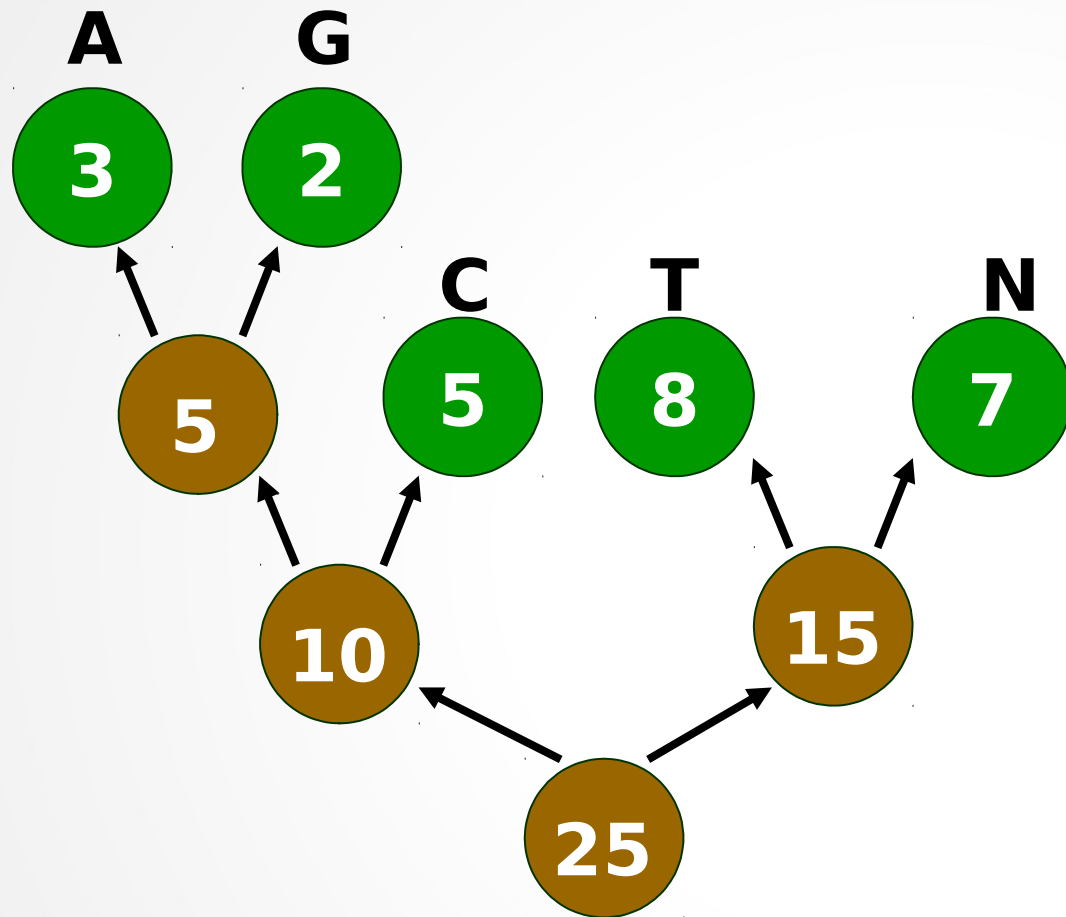
Exemple de création de l'arbre



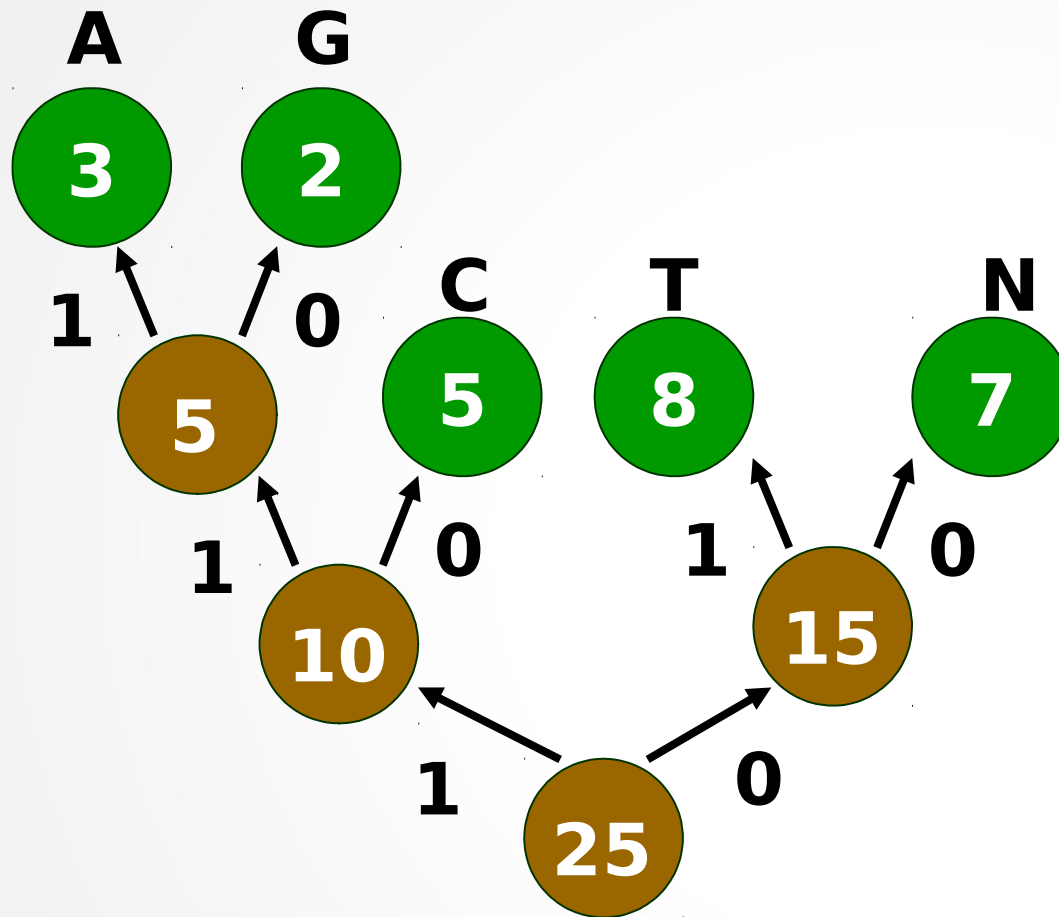
Exemple de création de l'arbre



Exemple de création de l'arbre

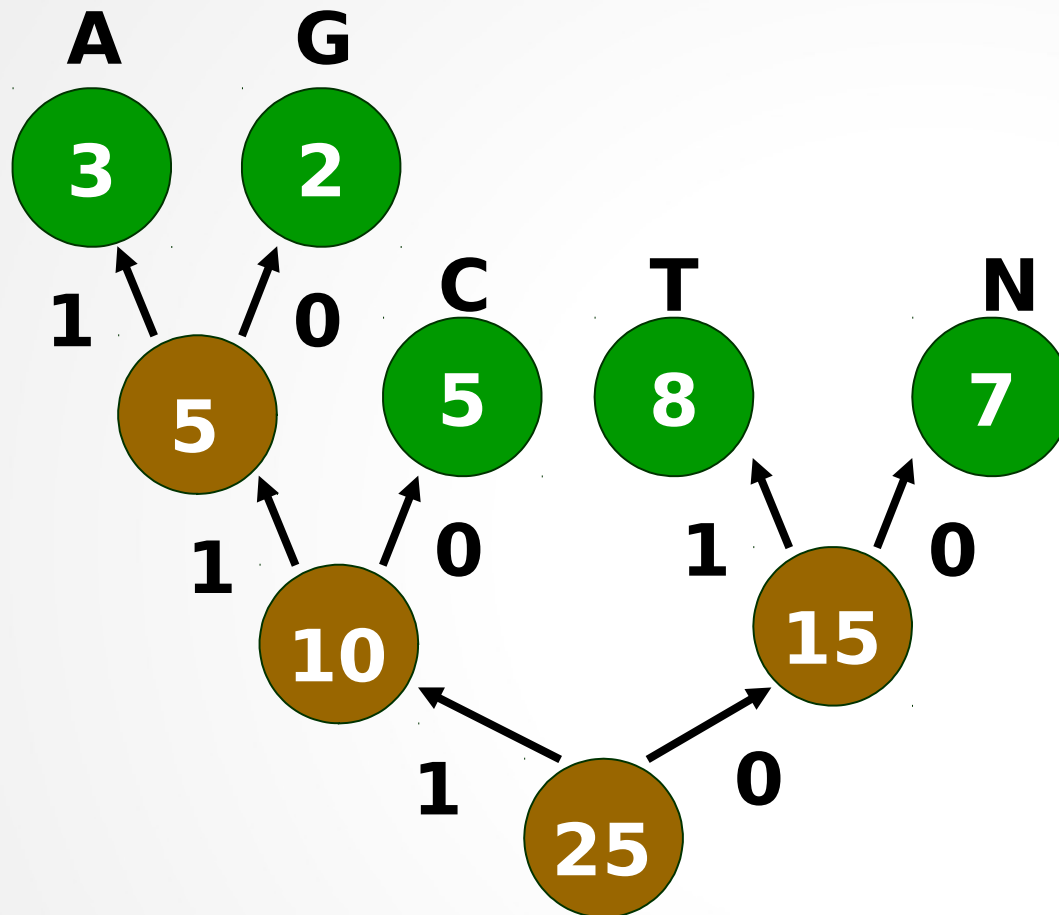


Exemple de création de l'arbre



Ajout de 0 ou 1 sur chaque branche
Suite de 0,1 unique entre la racine et une feuille

Exemple de création de l'arbre



T = 01

N = 00

C = 10

A = 111

G = 110

Compression d'Huffman

- L'arbre est utilisé sur chaque caractère de la séquence
- N N T N A C T T N G N N G T T N C C T A T A C C T
- 0000010011110010100110000011001010010100111101111101001

Compression d'Huffman

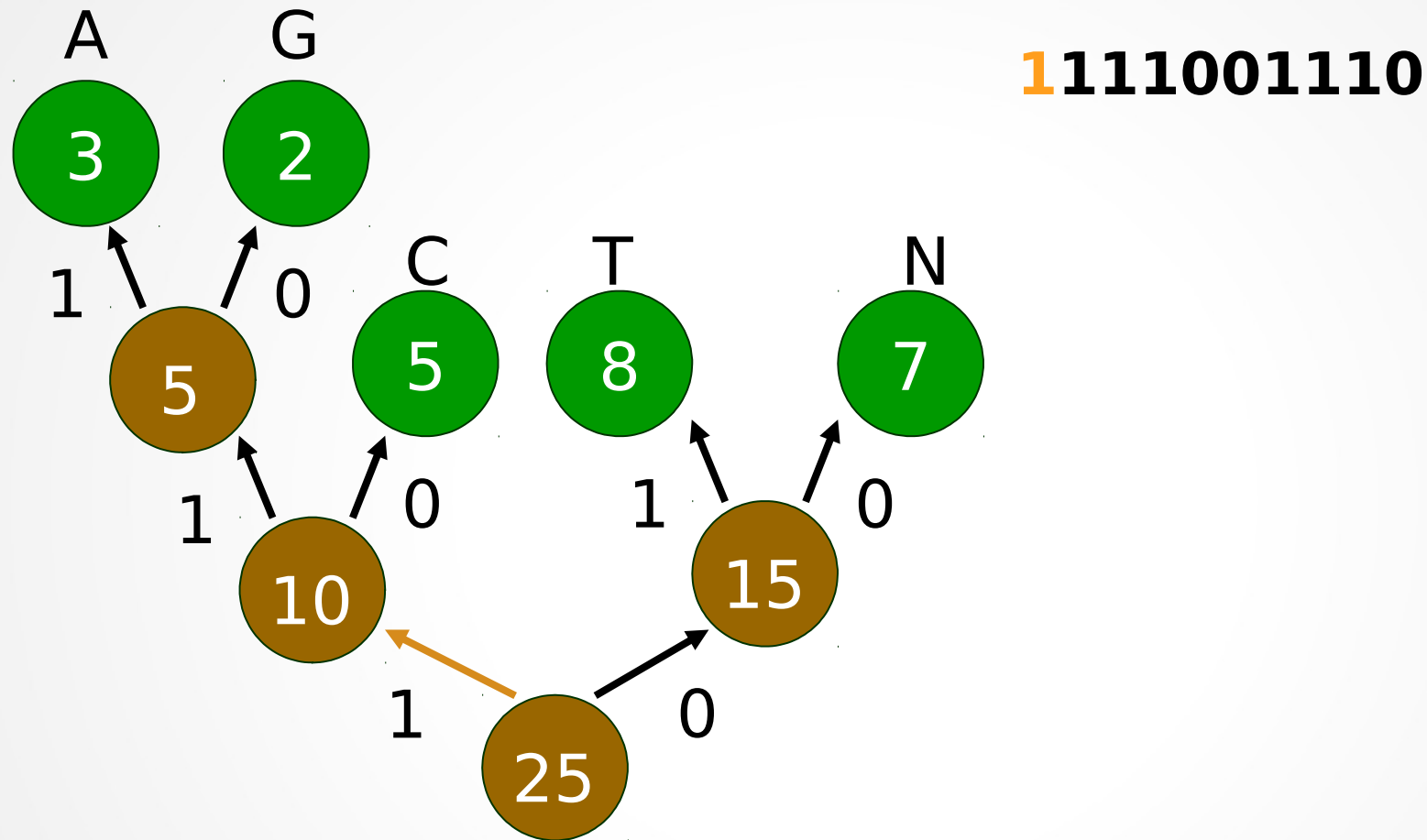
- L'arbre est utilisé sur chaque caractère de la séquence
- N N T N A C T T N G N N G T T N C C T A T A C C T
- 0000010011110010100110000011001010010100111101111101001
- Écrit la séquence par 8 bits (octet)
 - 00000100 = ''
 - 11110010 = 'ò'
 - 10011000 = 'Φ'
 - 00110010 = '2'
 - 10010100 = '☺'
 - 11110111 = '÷'
 - 1101001 = 'i'



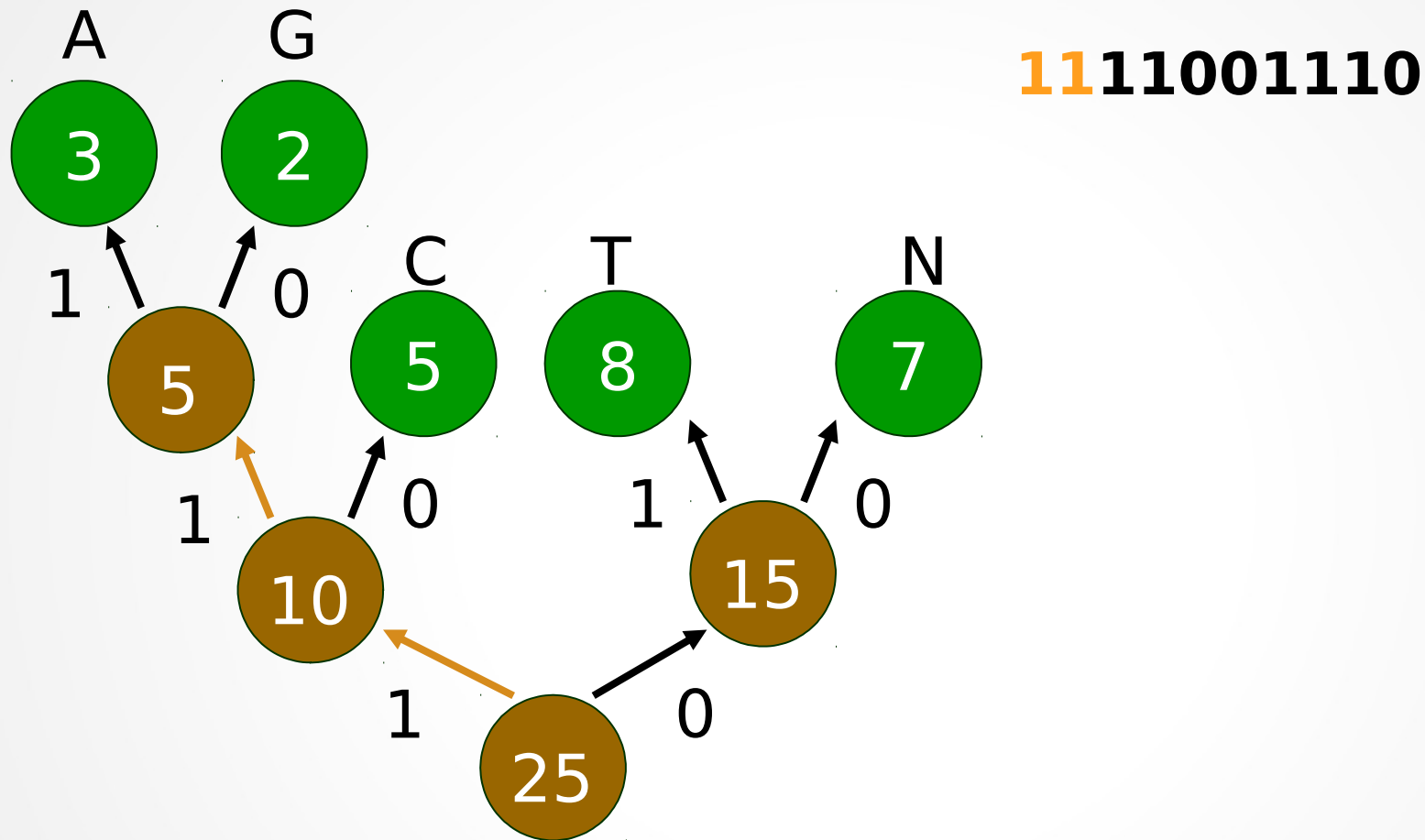
Algorithme simple de décompression d'Huffman

- Pour chaque caractère du fichier compressé
 - Décomposer les caractères en bits
 - Suivre le chemin de bit de l'arbre
 - Quand le chemin arrive a une feuille
 - Ecrire la lettre correspondante
 - Revenir a la racine de l'arbre

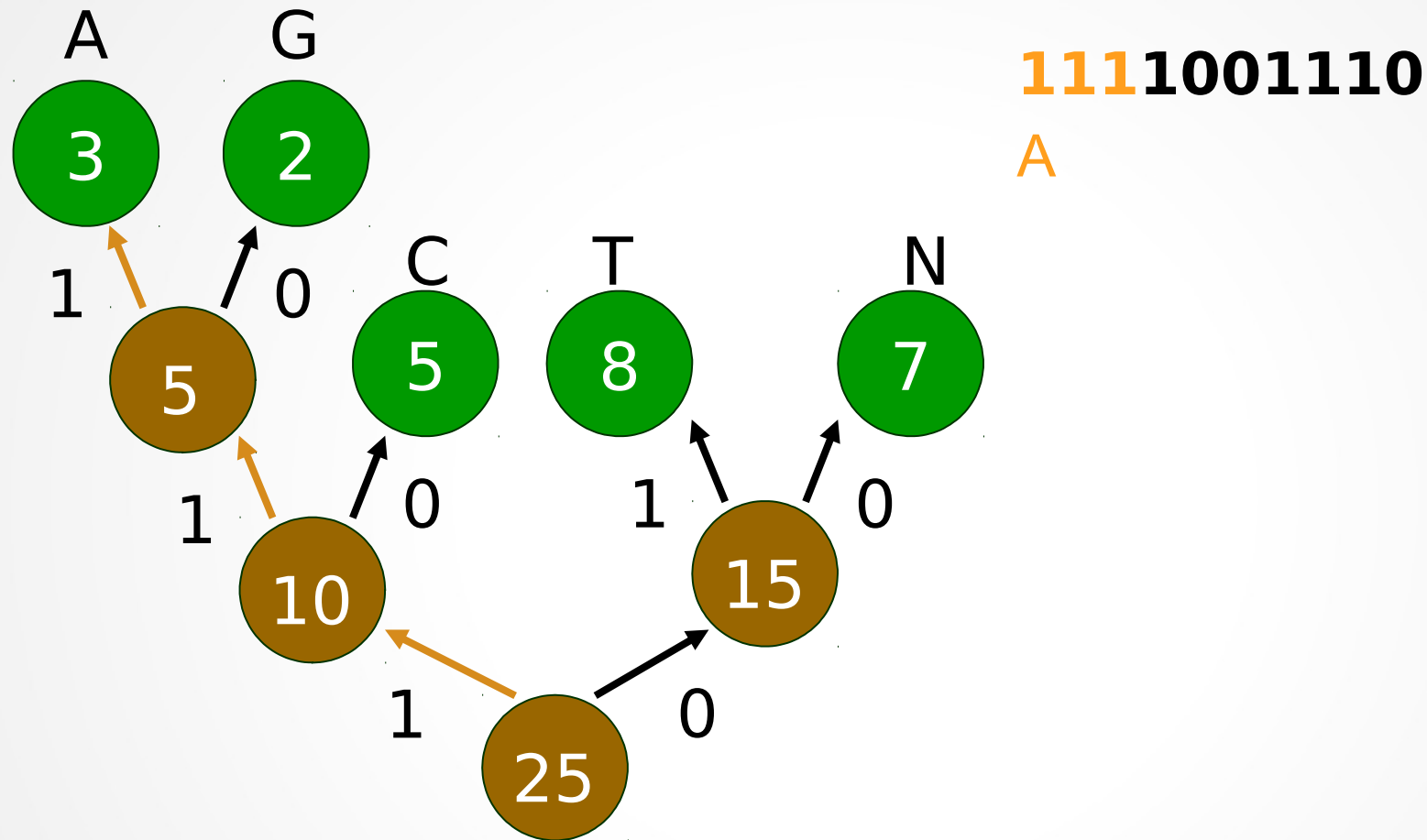
Exemple de décompression du texte



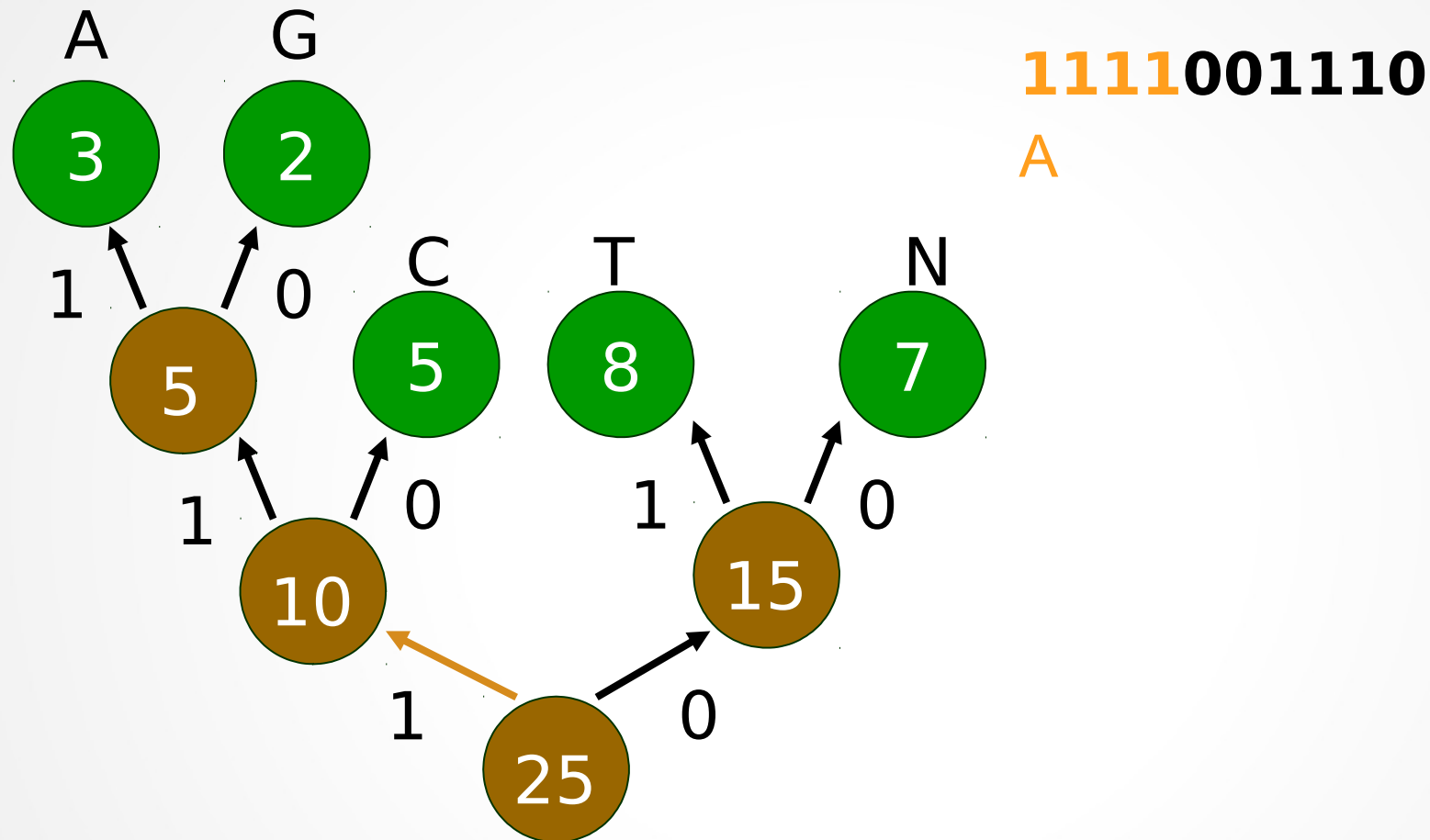
Exemple de décompression du texte



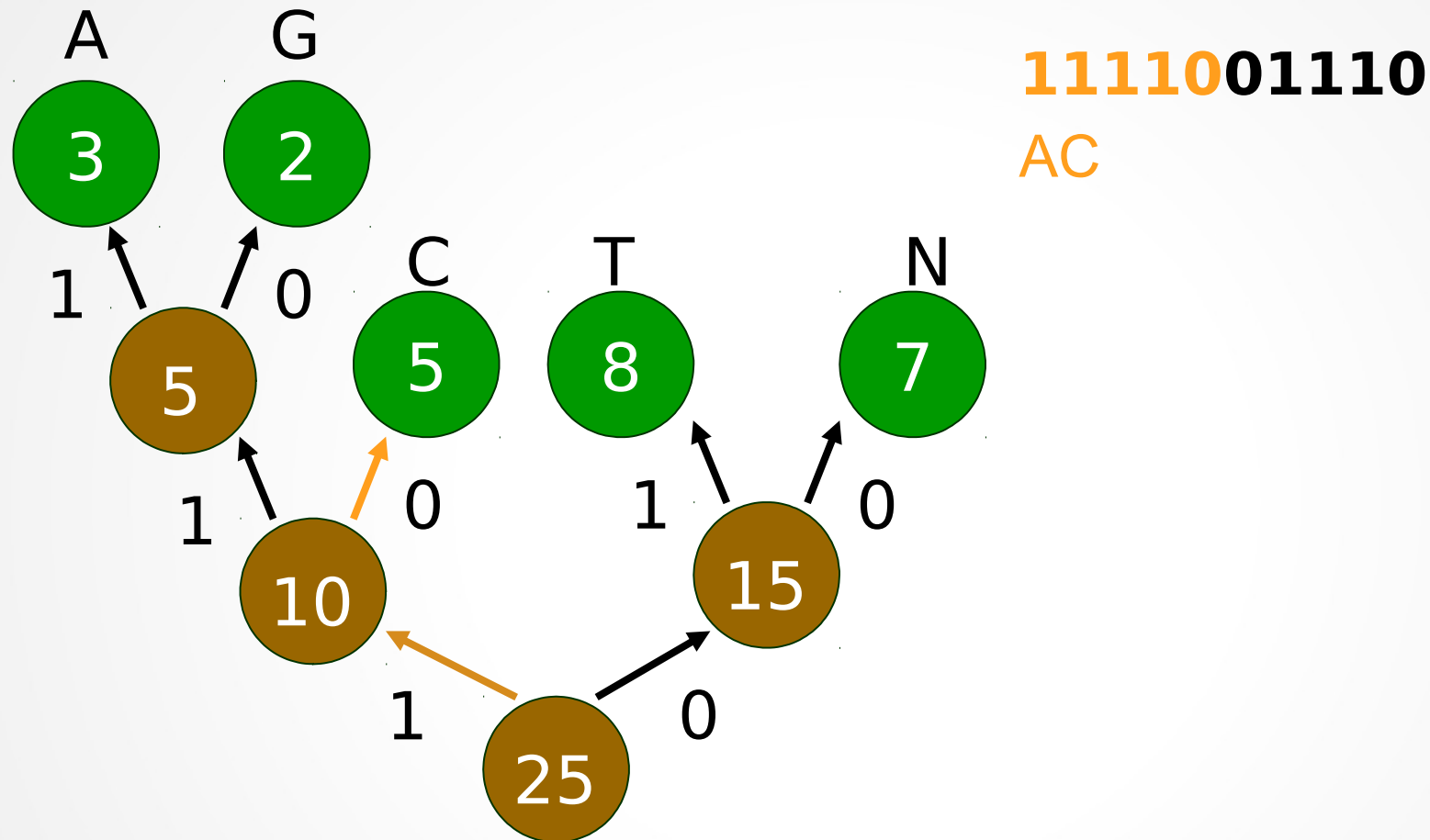
Exemple de décompression du texte



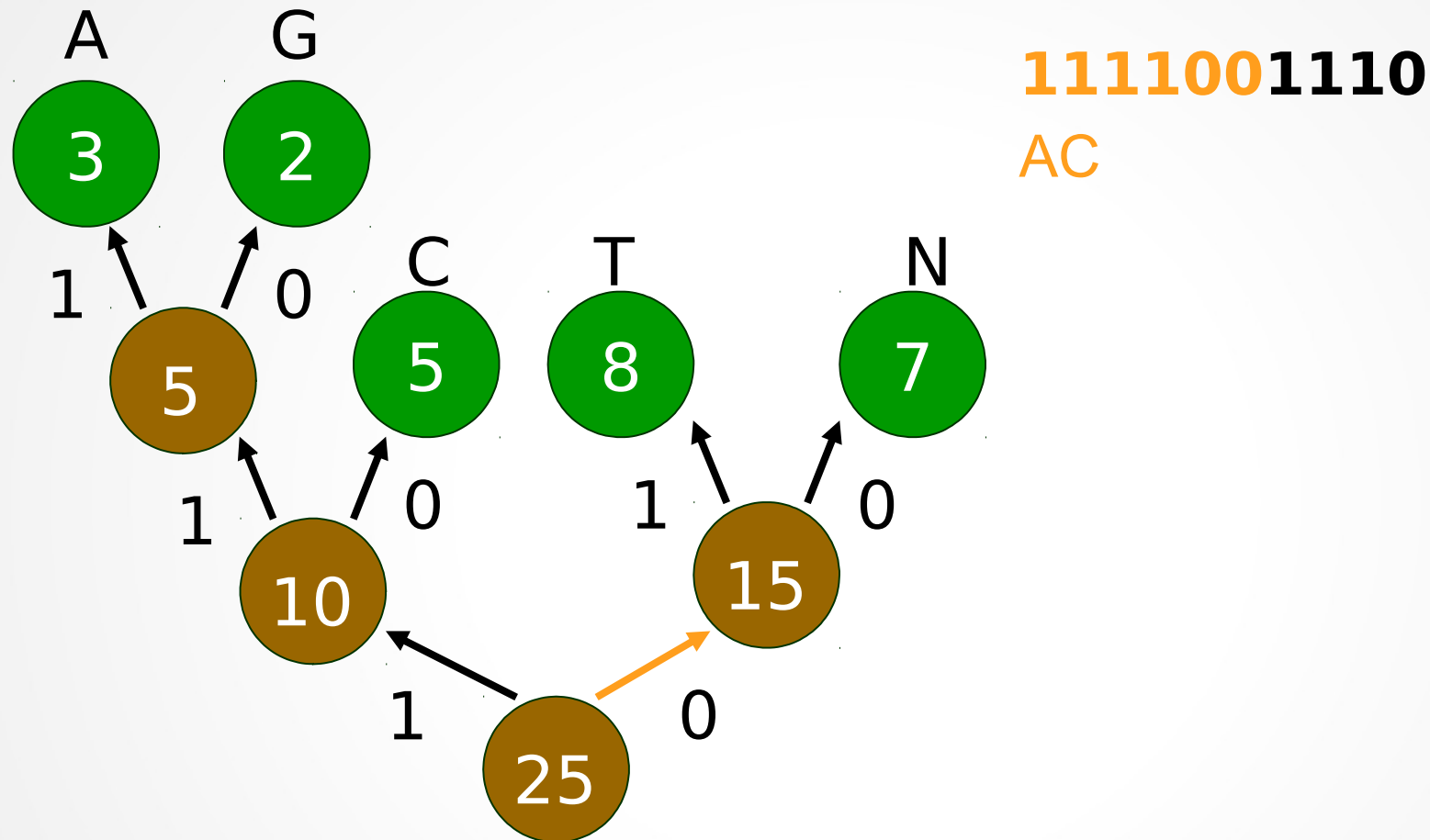
Exemple de décompression du texte



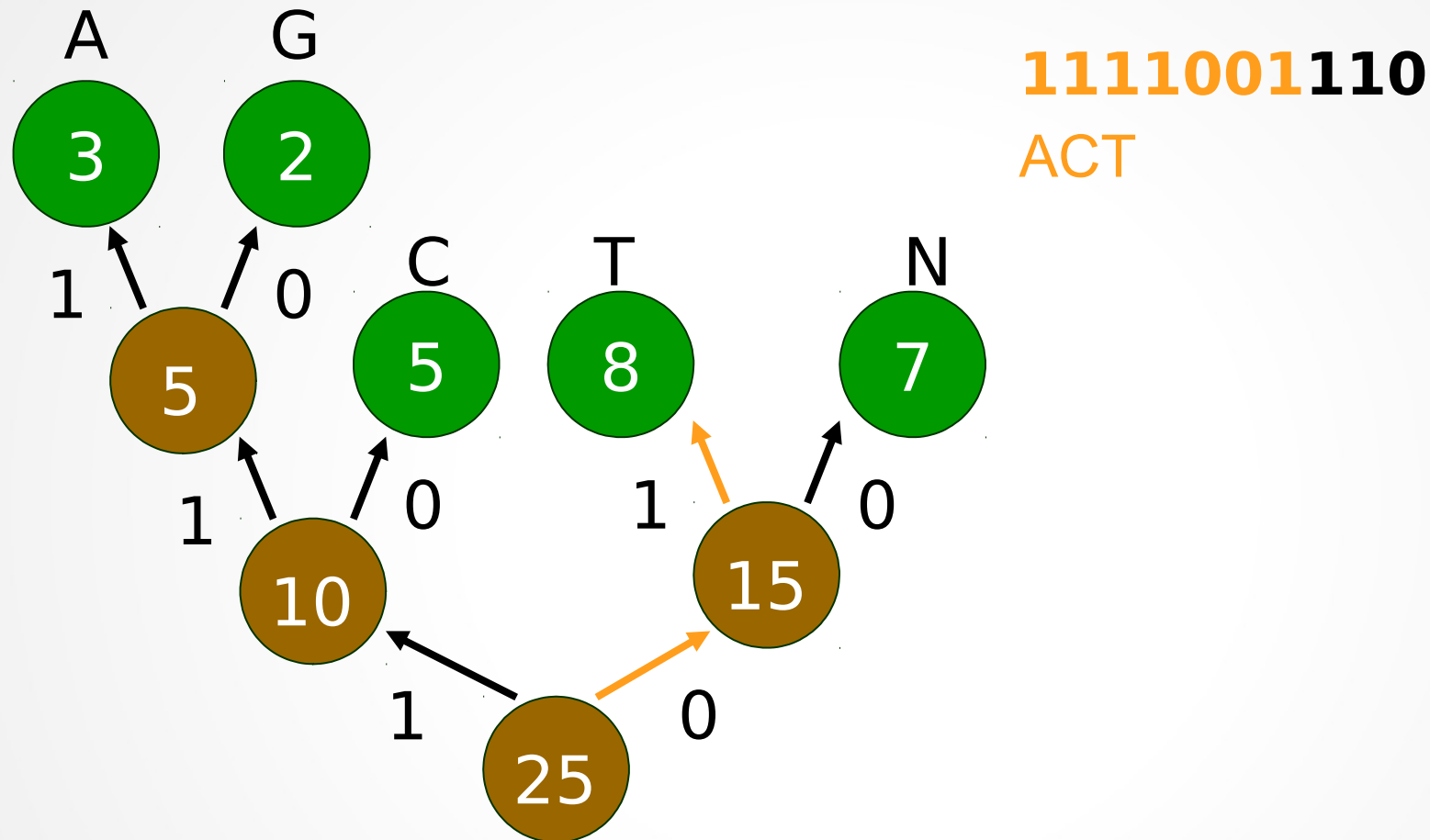
Exemple de décompression du texte



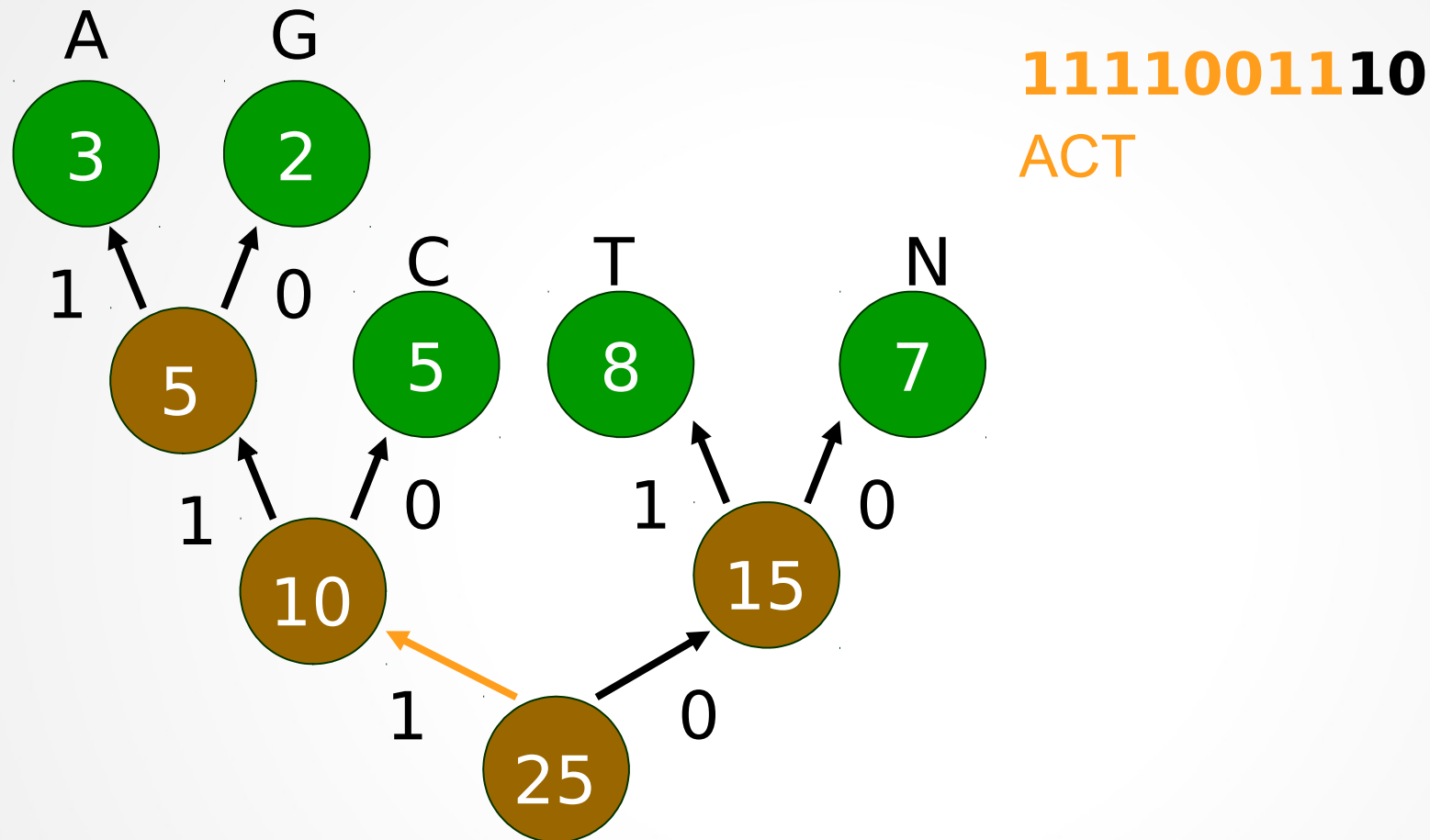
Exemple de décompression du texte



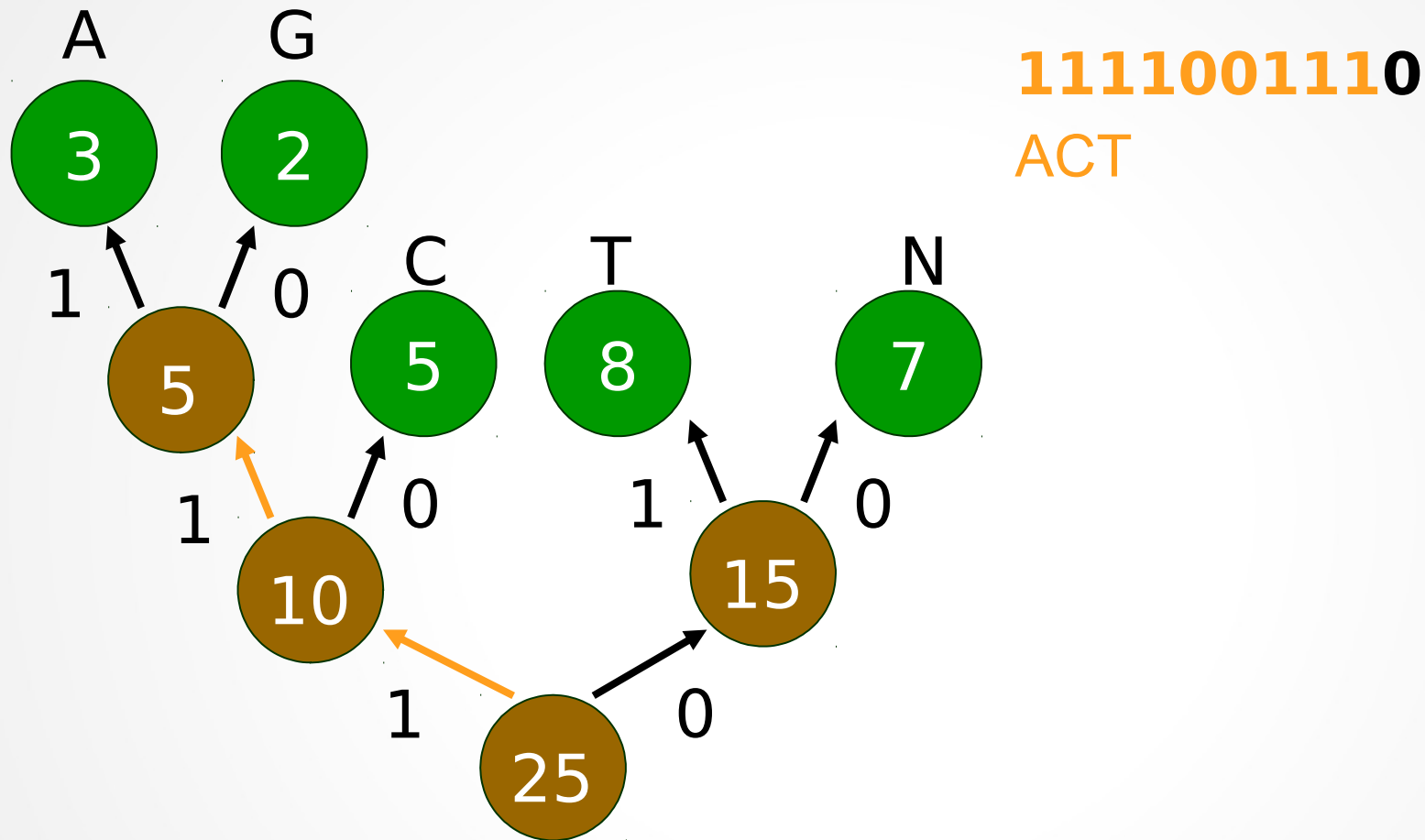
Exemple de décompression du texte



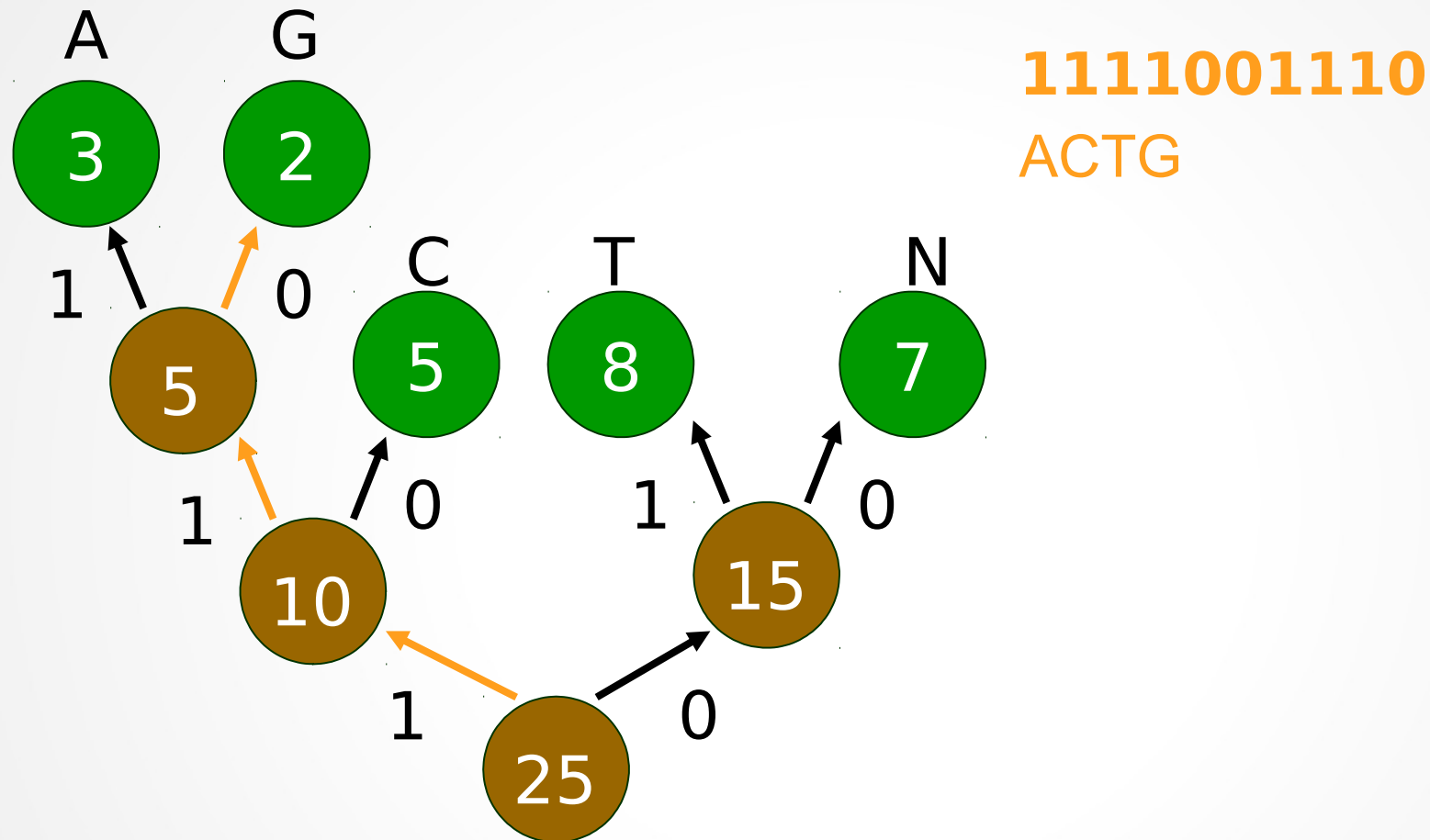
Exemple de décompression du texte



Exemple de décompression du texte



Exemple de décompression du texte



Projet a rendre

- Ecrire en langage python 1 script:
 - A partir d'une sequence d'ADN
 - Transformer la sequence avec BWT
 - Compresser la sequence
 - Sauvegarder la sequence (compresse ou non)
 - A partir d'une sequence d'ADN compresse
 - Decompresser la sequence
 - Transformer la sequence avec BWT
 - Sauvegarder la sequence
-
- Input: sequence d'ADN (A,C,G,N,T) simple sans header
- Fichier ReadMe pour l'utilisation du script

Et / Ou

Puis



Projet a rendre : bareme

- Style ecriture Python (4 pts)
 - (1) Code commente (~ autant de lignes de commentaires que de code)
 - (1) Aucune fonction de plus de 100 lignes
 - (1) Au moins 3 fichiers Python (main, BWT et Huffman)
 - (1) Variable avec des noms explicites
- Fonction BWT (6 pts)
 - (2) Transformation en sequence BWT
 - (2) Transformation de la BWT en sequence 'normale'
 - (2) Choix de l'affichage de la transformation
 - Affichage direct de la sequence BWT
 - Affichage de chaque etape de la transformation (via clavier)
- De/Compression Huffman (6 pts)
 - (1.5) Transformation et affichage de l'ADN en binaire
 - (1.5) Transformation et affichage du binaire en caractere
 - (1.5) Transformation et affichage des caracteres en binaire
 - (1.5) Transformation et affichage du binaire en ADN
- Interface graphique (4 pts)
 - (1) Ouverture et Sauvegarde d'un fichier
 - (2) Affichage sous forme de matrice de la (de)transformation BWT
 - (1) Affichage de Huffman
- Fonctionnalites supplementaires ou 'meilleur' algorithmique (4 pts)