

# Compression avec arbres quaternaires

Algorithmique et structures de données, 2022-2023

---

P. Albuquerque (B410), P. Künzli et O. Malaspinas (A401), ISC, HEPIA  
2023-05-09

En partie inspirés des supports de cours de P. Albuquerque

## Questions

- Structure de données d'un arbre quaternaire?

## Questions

- Structure de données d'un arbre quaternaire?

```
typedef struct _node {  
    int info;  
    struct _node *child[4];  
} node;
```

## Questions

- Structure de données d'un arbre quaternaire?

```
typedef struct _node {  
    int info;  
    struct _node *child[4];  
} node;
```

- Dessin d'un noeud d'arbre quaternaire (avec correspondance node)?

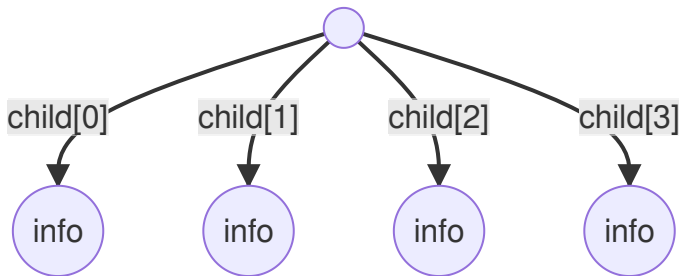
# Le cours précédent

## Questions

- Structure de données d'un arbre quaternaire?

```
typedef struct _node {  
    int info;  
    struct _node *child[4];  
} node;
```

- Dessin d'un noeud d'arbre quaternaire (avec correspondance node)?



## Questions

- Comment faire la symétrie d'axe horizontal?

## Questions

- Comment faire la symétrie d'axe horizontal?

```
arbre symétrie(arbre)
    si !est_feuille(arbre)
        échanger(arbre.enfant[0], arbre.enfant[2])
        échanger(arbre.enfant[1], arbre.enfant[4])
        pour i de 0 à 3
            symétrie(arbre.enfant[i])
    retourne arbre
```

# Compression sans perte

## Idée générale

- Regrouper les pixels par valeur

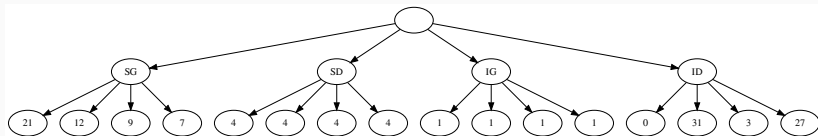
SG=0		SD=1		SG=0		SD=1
21		12		4		4
9		7		4		4
----- => -----						
1		1		0		31
1		1		3		27
IG=2		ID=3		IG=2		ID=3

- Comment faire?



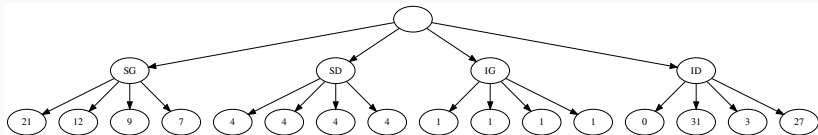
# Compression sans perte

Que devient l'arbre suivant?

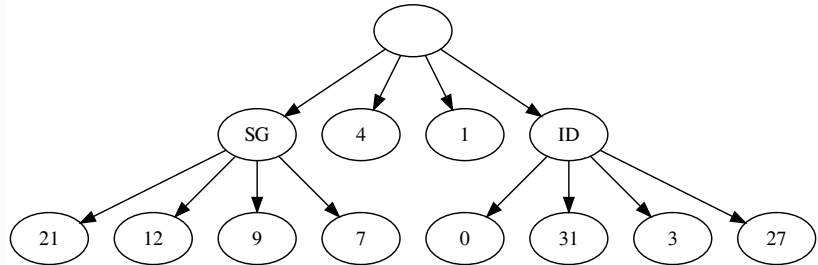


# Compression sans perte

Que devient l'arbre suivant?



Arbre compressé



# Compression sans perte

- Si un noeud a tous ses enfants égaux:
  - Donner la valeur au noeud,
  - Supprimer les enfants.
- Remonter jusqu'à la racine.

**Écrire le pseudo-code**

# Compression sans perte

- Si un noeud a tous ses enfants égaux:
  - Donner la valeur au noeud,
  - Supprimer les enfants.
- Remonter jusqu'à la racine.

## Écrire le pseudo-code

```
rien compression_sans_pertes(arbre)
  si !est_feuille(arbre)
    pour i de 0 à 3
      compression_sans_pertes(arbre.enfant[i])
    si derniere_branche(arbre)
      valeur, toutes_égales = valeur_enfants(arbre)
      si toutes_égales
        arbre.info = valeur
        detruire_enfants(arbre)
```

# Compression avec perte

## Idée générale

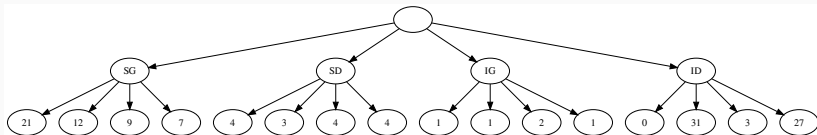
- Regrouper les pixels par valeur sous certaines conditions

SG=0   SD=1				SG=0   SD=1			
21		12		4		3	
9		7		4		4	
-----				=>	-----		
1		1		0		31	
2		1		3		27	
IG=2			ID=3		IG=2		ID=3

- On enlève si l'écart à la moyenne est "petit"?

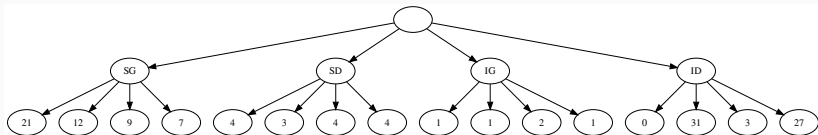
# Compression avec perte

Que devient l'arbre suivant si l'écart est petit?

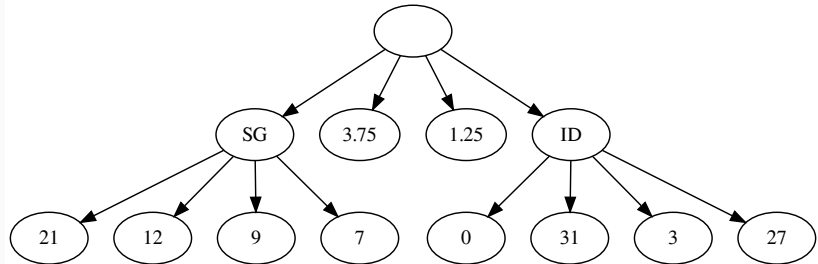


# Compression avec perte

Que devient l'arbre suivant si l'écart est petit?



Arbre compressé



Comment mesurer l'écart à la moyenne?



# Compression avec perte

## Comment mesurer l'écart à la moyenne?

- Avec l'écart-type

$$\mu = \frac{1}{4} \sum_{i=0}^3 p[i], \quad \sigma = \sqrt{\frac{1}{4} \sum_{i=0}^3 (\mu - p[i])^2} = \sqrt{\frac{1}{4} \left( \sum_{i=0}^3 p[i]^2 \right) - \mu^2}$$

## Que devient l'algorithme?

## Comment mesurer l'écart à la moyenne?

- Avec l'écart-type

$$\mu = \frac{1}{4} \sum_{i=0}^3 p[i], \quad \sigma = \sqrt{\frac{1}{4} \sum_{i=0}^3 (\mu - p[i])^2} = \sqrt{\frac{1}{4} \left( \sum_{i=0}^3 p[i]^2 \right) - \mu^2}$$

## Que devient l'algorithme?

- Si  $\sigma < \theta$ ,  $\theta$  est la **tolérance**:
  - Remplacer la valeur du pixel par la moyenne des enfants.
  - Remonter les valeurs dans l'arbre.

## Quelle influence de la valeur de $\theta$ sur la compression?

## Comment mesurer l'écart à la moyenne?

- Avec l'écart-type

$$\mu = \frac{1}{4} \sum_{i=0}^3 p[i], \quad \sigma = \sqrt{\frac{1}{4} \sum_{i=0}^3 (\mu - p[i])^2} = \sqrt{\frac{1}{4} \left( \sum_{i=0}^3 p[i]^2 \right) - \mu^2}$$

## Que devient l'algorithme?

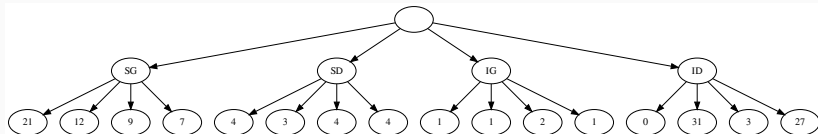
- Si  $\sigma < \theta$ ,  $\theta$  est la **tolérance**:
  - Remplacer la valeur du pixel par la moyenne des enfants.
  - Remonter les valeurs dans l'arbre.

## Quelle influence de la valeur de $\theta$ sur la compression?

- Plus  $\theta$  est grand, plus l'image sera compressée.

# Compression avec perte

Que devient l'arbre avec  $\theta = 0.5$ ?



**Figure 1:** L'arbre original.

# Compression avec perte

Que devient l'arbre avec  $\theta = 0.5$ ?

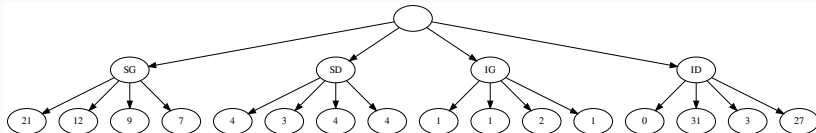


Figure 1: L'arbre original.

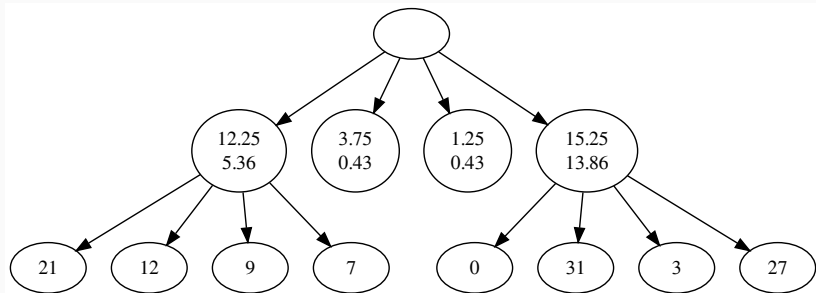


Figure 2: Arbre compressé.

**Modifications sur la structure de données?**

## Modifications sur la structure de données?

- On stocke la moyenne, et la moyenne des carrés.

```
struct noeud
    flottant moyenne, moyenne_carre
    node enfants[4]
```

- Comment on calcule moyenne et moyenne\_carre sur chaque noeud (pseudo-code)?

**Pseudo-code**



## Pseudo-code

```
rien moyenne(arbre) {  
    si !est_feuille(arbre)  
        pour enfant dans arbre.enfants  
            moyenne(enfant)  
    pour enfant dans arbre.enfants  
        arbre.moyenne += enfant.moyenne  
        arbre.moyenne_carre += enfant.moyenne_carre  
    arbre.moyenne /= 4  
    arbre.moyenne_carre /= 4
```

## Pseudo-code

# La compression avec pertes

## Pseudo-code

```
rien compression_avec_pertes(arbre, theta)
  si !est_feuille(arbre)
    pour i de 0 à 3
      compression_avec_pertes(arbre.enfant[i])
    si derniere_branche(arbre)
      si racine(arbre.moyenne_carre - arbre.moyenne^2) < theta
        detruire_enfants(arbre)
```

## Le code en entier

```
arbre = matrice_à_arbre(matrice)
moyenne(arbre)
compression_avec_pertes(arbre)
```