Polymorphisme

```
ArbreHuffman

Ø

virtual bool contient(char caracter) = 0;
virtual long getFrequence() const = 0;
...
virtual std::vector<bool> compresse(char caractere) = 0;
virtual char decompresse(std::vector<bool>* bits) = 0;
```

ArbreHuffmanBranche

```
-ArbreHuffman* gauche
-ArbreHuffman* droite
...
bool contient(char caracter) {
/* implémentation spécifique branches */
}
...
```

ArbreHuffmanFeuille

-char caracter

```
-long frequence
...
bool contient(char caracter) {
/* implémentation spécifique feuilles */
}
...
```

Codage du Huffman

- Un algorithme de compression de données; publié en 1952 par David Albert Huffman (un doctorant au MIT)
 - codage sans perte
 - code de taille variable
 - code court donnée aux symboles fréquents
 - aucun code n'est le préfix d'un autre
- Principe de codage
 - Un arbre composé de nœuds
 - Les feuilles contiennent les symboles à coder

"THIS IS AN EXAMPLE OF A HUFFMAN TREE"

Comptage du nombre d'occurrences de chaque symboles

T	Н	I	S	"	A	N	E	X	M	P	L	0	F	U	R
2	2	2	2	7	4	2	4	1	2	1	1	1	3	1	1

"	Α	E	F	T	Н	I	S	N	M	X	P	L	0	U	R
7	4	4	3	2	2	2	2	2	2	1	1	1	1	1	1

Chaque symbole devient un arbre (*ArbreHuffmanFeuille*)





















On regroupe les 2 arbres les moins fréquents (ArbreHuffmanBranche)













On trie les arbres

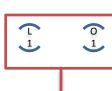












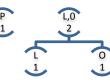












On trie les arbres

















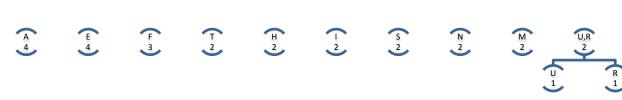
On regroupe les 2 arbres les moins fréquents

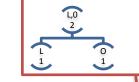






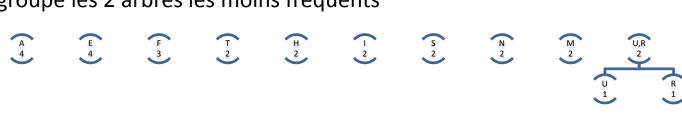


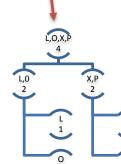




On trie les arbres (pas de changement)



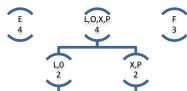




On trie les arbres





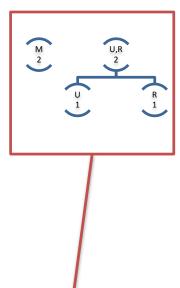






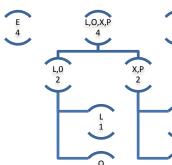








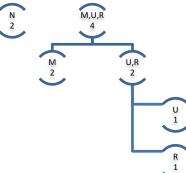








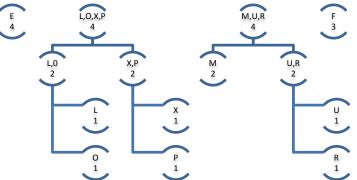




On trie les arbres



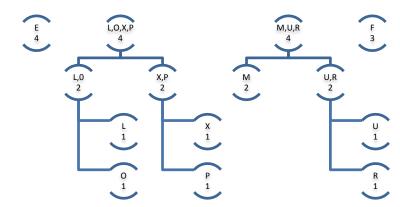




On regroupe les 2 arbres les moins fréquents



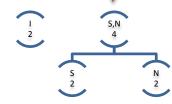




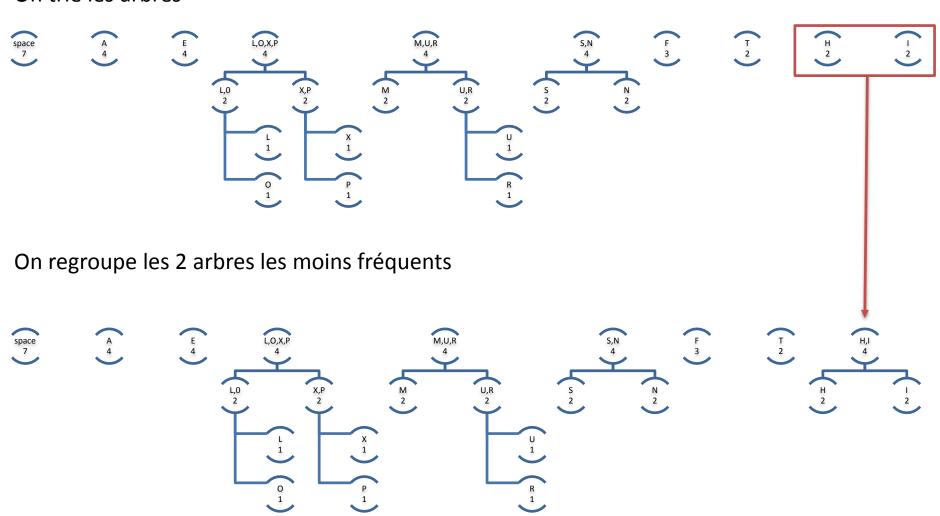


T H 2





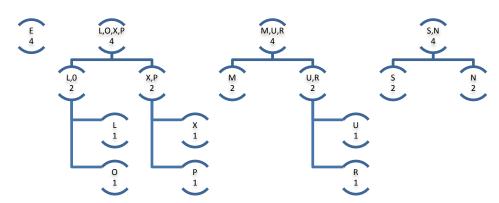
On trie les arbres



On trie les arbres

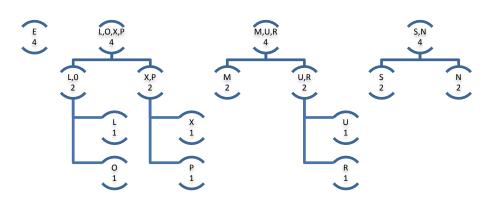


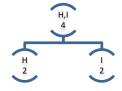


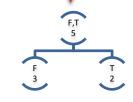




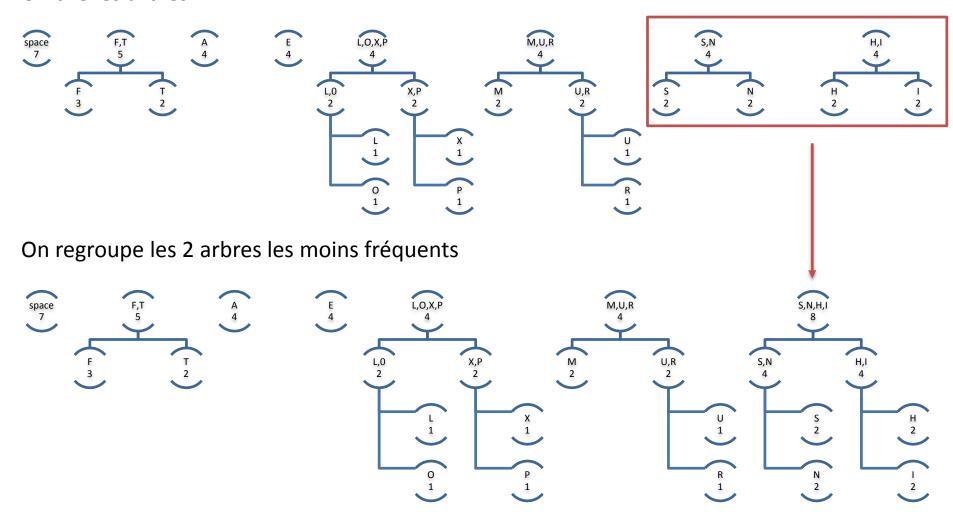




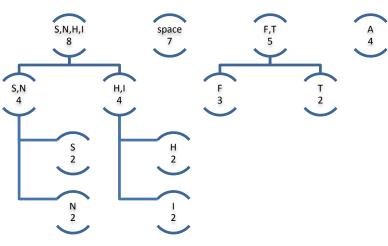


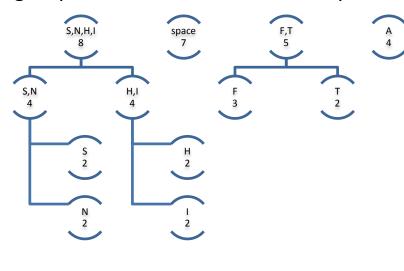


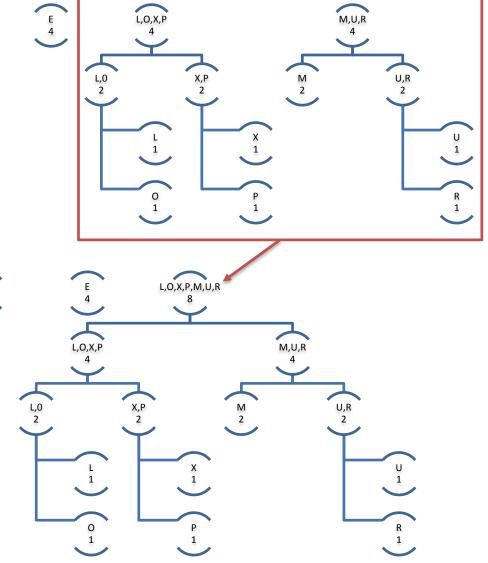
On trie les arbres



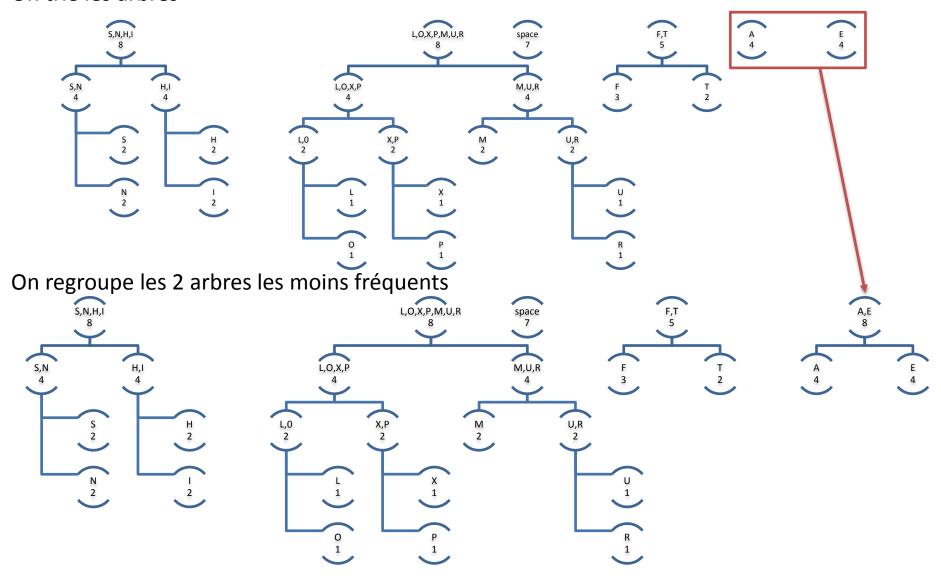
On trie les arbres

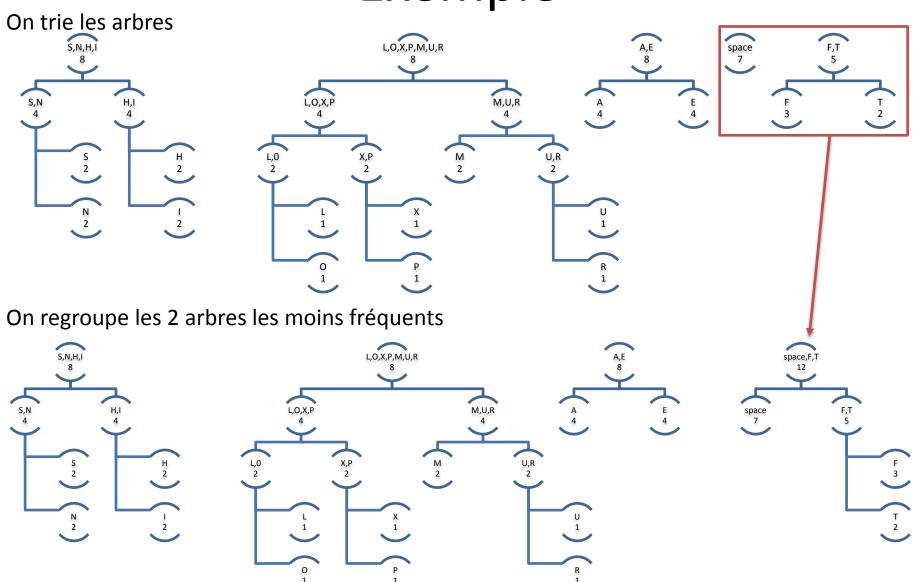




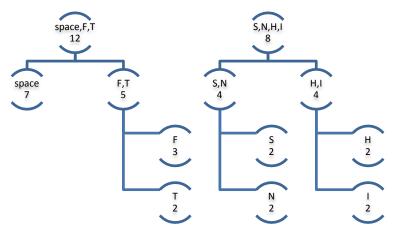


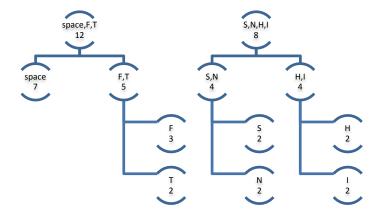
On trie les arbres

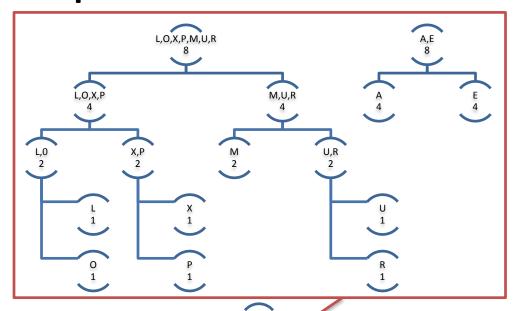


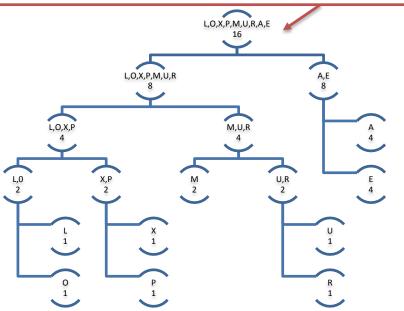


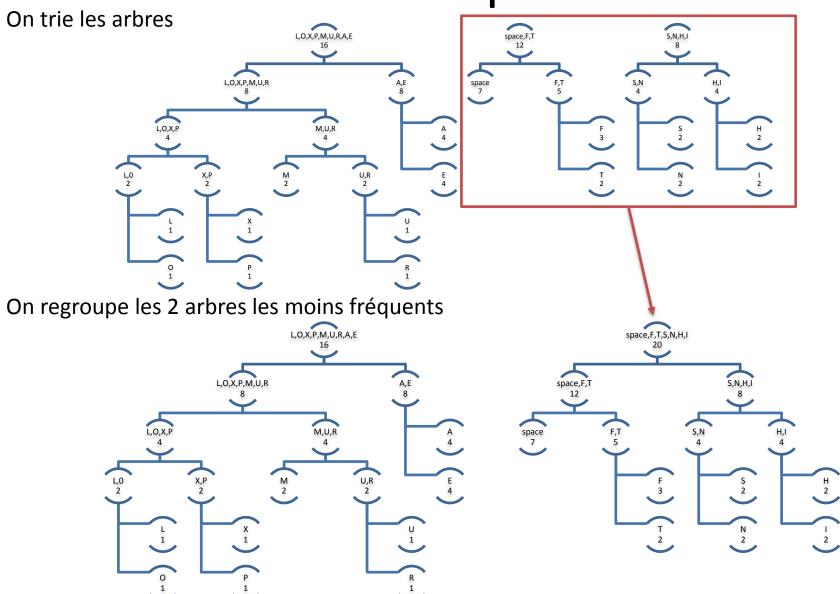
On trie les arbres



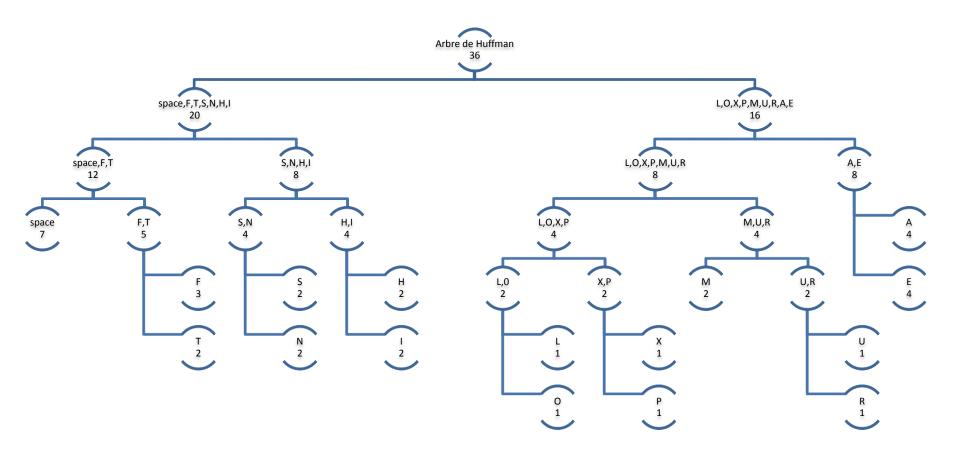








On trie les arbres et on regroupe les deux moins fréquents Tous les sous-arbres ont été regroupés, nous avons donc notre arbre de Huffman



Pour obtenir le code d'un symbole, on suite le chemin de la racine jusqu'à la feuille. Le chemin emprunté correspond au codage:

- 1 sous-arbre gauche;
- 0 sous-arbre droite.

