

## Travail pratique #2

### Compression par dictionnaire – Compression d'images

#### Questions

Donnez le développement pour vos réponses, en particulier quand le calcul est long.

1. (*30 points.*) **Compression par dictionnaire.** Dans les sous-questions qui suivent, on considère que LZ77 (et sa variante LZSS) respecte les conventions suivantes.

**Convention I:** LZ77 recherche toujours une copie ayant la longueur  $l$  la plus grande possible.

**Convention II:** lorsqu'il existe plusieurs copies de longueur maximale, considérez que LZ77 favorise celle qui a le décalage  $o$  le plus petit possible.

**Convention III:** le dernier triplet de LZ77 pourrait ne pas respecter les conventions I et II, ceci parce que le texte d'entrée se termine. La priorité est de produire une suite de triplets qui décrivent exactement le texte d'entrée.

Dans les sous-questions (a), (b), (c) et (d), considérez l'entrée suivante.

**acbcbadbcdbaacdbbcacbdda**

- (a) Donnez la suite de triplets  $\langle o, l, c \rangle$  qui sont produits en traitant l'entrée à l'aide de la technique LZ77 de base (présentée à partir de la page 10 des acétates). Considérez que ni la fenêtre d'inspection ni la fenêtre de recherche sont limitées en taille. Lorsqu'il n'y a pas de copie trouvée, i.e.  $l = 0$ , fixez arbitrairement le décalage à zéro, i.e.  $o = 0$ .
- (b) Donnez la suite des messages produits en traitant l'entrée à l'aide de LZSS (présentée à la page 17 des acétates). Un message est soit une paire  $\langle o, l \rangle$ , lorsqu'une copie est trouvée, soit un singleton  $\langle c \rangle$ , autrement. Considérez que LZSS ne produit une paire que lorsque la copie est non-triviale, i.e. telle que  $l \geq 2$ . Le reste du comportement de LZSS est similaire à celui de LZ77.
- (c) Décrivez le traitement de l'entrée par LZ78 (présentée à partir de la page 18 des acétates), en montrant l'état du dictionnaire à chaque étape. Considérez que le dictionnaire contient initialement une seule entrée: l'entrée 0 associée à la chaîne  $\epsilon$ .

(d) Décrivez le traitement de l'entrée par LZW (présentée à partir de la page 32 des acétates), en montrant l'état du dictionnaire à chaque étape. Considérez que l'alphabet est  $\{\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}\}$ ; i.e. initialisez le dictionnaire avec les chaînes  $\mathbf{a}$ ,  $\mathbf{b}$ ,  $\mathbf{c}$  et  $\mathbf{d}$  (dans l'ordre).

(e) Décodez la séquence de triplets suivante à la façon de LZ77.

$\langle 0, 0, \mathbf{a} \rangle, \langle 0, 0, \mathbf{c} \rangle, \langle 0, 0, \mathbf{d} \rangle, \langle 0, 0, \mathbf{b} \rangle, \langle 1, 2, \mathbf{a} \rangle, \langle 6, 1, \mathbf{a} \rangle, \langle 0, 0, \mathbf{c} \rangle$

(f) Décodez la séquence de messages suivante à la façon de LZSS.

$\langle \mathbf{a} \rangle, \langle \mathbf{b} \rangle, \langle \mathbf{b} \rangle, \langle \mathbf{c} \rangle, \langle \mathbf{d} \rangle, \langle \mathbf{c} \rangle, \langle 6, 2 \rangle, \langle 5, 2 \rangle$

(g) Décodez la séquence de couples suivante à la façon de LZ78.

$\langle 0, \mathbf{b} \rangle, \langle 0, \mathbf{a} \rangle, \langle 0, \mathbf{d} \rangle, \langle 1, \mathbf{c} \rangle, \langle 2, \mathbf{b} \rangle, \langle 0, \mathbf{c} \rangle, \langle 3, \mathbf{b} \rangle$

(h) Décodez la séquence d'index suivante à la façon de LZW. Considérez que le dictionnaire est initialisé avec les chaînes  $\mathbf{a}$ ,  $\mathbf{b}$ ,  $\mathbf{c}$  et  $\mathbf{d}$  (dans l'ordre).

1, 2, 4, 3, 3, 4, 5, 9

(i) Si on refaisait un exercice comme la question 1(b) mais en cessant de respecter la Convention II, on tomberait sur des cas où plusieurs plus longues copies sont disponibles. Énumérez toutes les façons dont LZSS pourrait traiter l'entrée suivante en respectant la Convention I mais pas la Convention II. Notez qu'on continue à restreindre les copies à être d'une longueur d'au moins deux ( $l \geq 2$ ). Permettez les copies telles que  $l > o$ . (*Indice*: vous devriez trouver 2 façons distinctes.)

### lebardubarbarebarbu

(j) Maintenant, si on refaisait la question 1(i) mais sans respecter la Convention I non plus, on observerait une multiplication des façons différentes de traiter l'entrée. Énumérez toutes les façons dont LZSS pourrait traiter l'entrée suivante. Notez qu'il n'est même pas nécessaire de préférer une copie à un singleton. Notez qu'on continue à restreindre les copies à être d'une longueur d'au moins deux ( $l \geq 2$ ). (*Indice*: vous devriez trouver 4 façons distinctes.)

### raisinraide

2. (25 points.) **Prédiction basée sur les contextes.** Ici, nous considérons la variante PPM A. Supposons que le début de texte `yokohamaokayam` a déjà été prédit et encodé. Les tables suivantes présentent les modèles de Markov d'ordres  $-1$ ,  $0$ ,  $1$ ,  $2$  et  $3$  cumulés à ce point.

Vous devez indiquer comment procéderaient les traitements indiqués dans les sous-questions. À chaque sous-question, il faut partir de l'état décrit dans les tables; i.e. c'est comme si, à chaque sous-question, on envisageait un scénario différent à la suite de `yokohamaokayam`. Il ne faut pas oublier de mettre à jour les tables après l'encodage de chaque symbole.

Décrivez la(les) opération(s) successive(s) effectuée(s) lors du traitement, comme: "échappement vers le modèle d'ordre  $k$ ", "encodage du symbole  $c$  dans le modèle d'ordre  $k$  avec probabilité  $p$ ", "incrémentation du poids du symbole  $c$  dans le contexte  $w$ ", etc.

Ordre -1		
C	S	N
—	a	1
	b	1
	c	1
	d	1
	e	1
	f	1
	g	1
	h	1
	i	1
	j	1
	k	1
	m	1
	n	1
	o	1
	p	1
	r	1
	s	1
	t	1
	u	1
	w	1
	y	1
	z	1

Ordre 0		
C	S	N
—	a	4
	h	1
	k	2
	m	2
	o	3
	y	2
	<Esc>	1

Ordre 1		
C	S	N
a	m	2
	o	1
	y	1
	<Esc>	1
h	a	1
	<Esc>	1
k	a	1
	o	1
	<Esc>	1
m	a	1
	<Esc>	1
o	h	1
	k	2
	<Esc>	1
y	a	1
	o	1
	<Esc>	1

Ordre 2		
C	S	N
am	a	1
	<Esc>	1
ao	k	1
	<Esc>	1
ay	a	1
	<Esc>	1
ha	m	1
	<Esc>	1
ka	y	1
	<Esc>	1
ko	h	1
	<Esc>	1
ma	o	1
	<Esc>	1
oh	a	1
	<Esc>	1
ok	a	1
	o	1
	<Esc>	1
ya	m	1
	<Esc>	1
yo	k	1
	<Esc>	1

- Le prochain symbole est `a`.
- Le prochain symbole est `i`.
- Le prochain symbole est `o`.
- Refaire les trois traitements précédents mais en utilisant le principe d'exclusion.

3. (15 points.) **Transformées de Burrows-Wheeler et “Move-to-front”.**
- (a) Effectuez la BWT sur la chaîne `well_is_this_well_well`. N’oubliez pas de calculer aussi le rang de la chaîne.
  - (b) Défaites la BWT sur `(nrsoocimpse, 0)`.
  - (c) Effectuez la transformée “Move-to-front” sur `nrsoocimpse` en prenant comme liste initiale `c, e, i, m, n, o, p, r, s`, dans l’ordre.

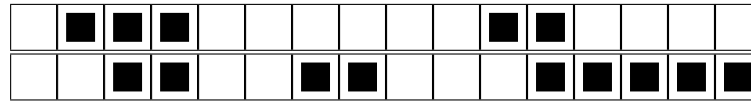
4. (20 points.) **Compression d’images sans perte JPEG.** À la page 5 des acétates sur la compression d’images sans perte, huit prédicteurs sont présentés. Testez empiriquement ces huit prédicteurs sur l’image test qui accompagne cet énoncé de travail pratique. L’image est du format PGM (*Portable GrayMap*), en version texte. Le format PGM est archi-simple. Il contient, dans l’ordre:

- la signature, “P2”;
- la largeur  $L$  de l’image, en pixels; ici, 512;
- la hauteur  $H$  de l’image, en pixels; ici, 512;
- l’intensité maximale d’un pixel; i.e. celle qui correspond à du blanc; ici, 255;
- les  $H * L$  pixels constituant l’image.

Chaque morceau d’information (chaque valeur) est séparé de ces voisins par des espaces blancs. À cause de la simplicité du format PGM, le traitement de l’image dans votre langage de programmation préféré devrait être des plus faciles. Pour les fins de la prédiction, adoptez les conventions suivantes:

- le premier prédicteur prédit systématiquement que la valeur d’un pixel est 128;
- chez tous les autres prédicteurs, lorsqu’ils doivent accéder à un pixel voisin qui n’existe pas (à cause des limites de l’image), supposez que le pixel voisin a l’intensité 128;
- calculez l’erreur de prédiction pour un pixel comme étant la valeur absolue entre l’intensité prédite et l’intensité réelle; i.e.  $|\hat{I}(i, j) - I(i, j)|$ ;
- l’erreur de prédiction pour l’image au complet est la somme des erreurs de prédiction pour les pixels individuels;
- le meilleur prédicteur est celui qui minimise l’erreur de prédiction pour l’image au complet.

5. (10 points.) **Encodage de facsimile.** La grille qui suit présente deux lignes de pixels consécutives sur une image binaire. La seconde ligne doit être encodée de deux façons différentes. Dans les deux cas, fournissez le développement ainsi que les mots de code générés par l'encodage.



- (a) Décrivez l'encodage de la seconde ligne à l'aide de la technique unidimensionnelle, présentée aux pages 51 à 54 des acétates.
- (b) Décrivez l'encodage de la seconde ligne à l'aide de la technique bidimensionnelle, présentée aux pages 55 à-63.

## Remise des travaux

Vous devez remettre le travail via **monPortail**. La date de remise est repoussée au **18 avril**. Les autres modalités de remise sont inscrites dans le plan de cours.