TD d'algorithmique – TD5 – Images en noir et blanc

Le but de ce tableau est de manipuler des tableaux de tableaux de booléens qui représentent des images en noir et blanc. En effet, on va coder une image en noir et blanc de la manière suivante : chaque pixel noir correspond au booléen FAUX (pixel éteint) et chaque pixel blanc au booléen VRAI (pixel allumé). Un tableau de booléens correspond donc à une colonne de pixels, et un tableau de tableaux de booléens correspond à une image. Par exemple, l'image ci-contre correspond au tableau {{VRAI,VRAI,VRAI},{VRAI,VRAI,FAUX},{FAUX,FAUX,VRAI}}.

• Q1. Dessinez l'image qui correspond au tableau {{VRAI,VRAI},{FAUX,VRAI},{VRAI,FAUX}}.

• Q2. Écrivez l'instruction en pseudo-code qui permet de stocker dans une variable *zigzag* le tableau de tableaux de booléens correspondant à l'image suivante :  ?

• Q3. Quelle instruction en pseudo-code stocke dans une variable *PixelEnBasADroite* la valeur (VRAI ou FAUX) correspondant à la couleur (blanc ou noir) du pixel en bas à droite de *zigzag* ?

TD d'algorithmique – TD5 – Images en noir et blanc

Le but de ce tableau est de manipuler des tableaux de tableaux de booléens qui représentent des images en noir et blanc. En effet, on va coder une image en noir et blanc de la manière suivante : chaque pixel noir correspond au booléen FAUX (pixel éteint) et chaque pixel blanc au booléen VRAI (pixel allumé). Un tableau de booléens correspond donc à une colonne de pixels, et un tableau de tableaux de booléens correspond à une image. Par exemple, l'image ci-contre correspond au tableau {{VRAI,VRAI,VRAI},{VRAI,VRAI,FAUX},{FAUX,FAUX,VRAI}}.

• Q1. Dessinez l'image qui correspond au tableau {{VRAI,VRAI},{FAUX,VRAI},{VRAI,FAUX}}.

• Q2. Écrivez l'instruction en pseudo-code qui permet de stocker dans une variable *zigzag* le tableau de tableaux de booléens correspondant à l'image suivante :  ?

• Q3. Quelle instruction en pseudo-code stocke dans une variable *PixelEnBasADroite* la valeur (VRAI ou FAUX) correspondant à la couleur (blanc ou noir) du pixel en bas à droite de *zigzag* ?

TD d'algorithmique – TD5 – Images en noir et blanc

Le but de ce tableau est de manipuler des tableaux de tableaux de booléens qui représentent des images en noir et blanc. En effet, on va coder une image en noir et blanc de la manière suivante : chaque pixel noir correspond au booléen FAUX (pixel éteint) et chaque pixel blanc au booléen VRAI (pixel allumé). Un tableau de booléens correspond donc à une colonne de pixels, et un tableau de tableaux de booléens correspond à une image. Par exemple, l'image ci-contre correspond au tableau {{VRAI,VRAI,VRAI},{VRAI,VRAI,FAUX},{FAUX,FAUX,VRAI}}.

• Q1. Dessinez l'image qui correspond au tableau {{VRAI,VRAI},{FAUX,VRAI},{VRAI,FAUX}}.

• Q2. Écrivez l'instruction en pseudo-code qui permet de stocker dans une variable *zigzag* le tableau de tableaux de booléens correspondant à l'image suivante :  ?

• Q3. Quelle instruction en pseudo-code stocke dans une variable *PixelEnBasADroite* la valeur (VRAI ou FAUX) correspondant à la couleur (blanc ou noir) du pixel en bas à droite de *zigzag* ?

TD d'algorithmique – TD5 – Images en noir et blanc

Le but de ce tableau est de manipuler des tableaux de tableaux de booléens qui représentent des images en noir et blanc. En effet, on va coder une image en noir et blanc de la manière suivante : chaque pixel noir correspond au booléen FAUX (pixel éteint) et chaque pixel blanc au booléen VRAI (pixel allumé). Un tableau de booléens correspond donc à une colonne de pixels, et un tableau de tableaux de booléens correspond à une image. Par exemple, l'image ci-contre correspond au tableau {{VRAI,VRAI,VRAI},{VRAI,VRAI,FAUX},{FAUX,FAUX,VRAI}}.

• Q1. Dessinez l'image qui correspond au tableau {{VRAI,VRAI},{FAUX,VRAI},{VRAI,FAUX}}.

• Q2. Écrivez l'instruction en pseudo-code qui permet de stocker dans une variable *zigzag* le tableau de tableaux de booléens correspondant à l'image suivante :  ?

• Q3. Quelle instruction en pseudo-code stocke dans une variable *PixelEnBasADroite* la valeur (VRAI ou FAUX) correspondant à la couleur (blanc ou noir) du pixel en bas à droite de *zigzag* ?

- Q4. Écrivez un algorithme **LargeurImage** qui prend en entrée un tableau de tableaux de booléens codant une image en noir et blanc et qui renvoie la largeur de l'image correspondante. Par exemple, **LargeurImage**(*zigzag*) renvoie 4.
- Q4'. Même question pour l'algorithme **HauteurImage** (bien sûr **HauteurImage**(*zigzag*) renvoie 2).
- Q5. Écrivez un algorithme **CompteBlancsColonne** qui prend en entrée un tableau de booléens et renvoie le nombre de cases contenant VRAI dans ce tableau. Par exemple, **CompteBlancsColonne**({VRAI,FAUX,VRAI,VRAI}) renvoie 3.
- Q6. Sans faire de boucle (en utilisant un appel de l'algorithme **CompteBlancsColonne**), écrivez un algorithme **CompteNoirsColonne** qui prend en entrée un tableau de booléens et renvoie son nombre de cases contenant FAUX.
- Q7. Écrivez un algorithme **CompteBlancs** qui prend en entrée un tableau de tableaux de booléens codant une image et renvoie le nombre de pixels blancs dans l'image codée par ce tableau de tableau de booléens.
- Q8. On dispose d'un système qui permet d'afficher un pixel blanc pour 0.2€ par heure, et d'afficher un pixel noir pour 0.1€ par heure. Écrivez un algorithme **CoutAffichage** qui prend en entrée un tableau de tableaux de booléens *tabImage* codant une image et deux entiers *h* et *m*, et renvoie le coût d'affichage de l'image pendant *h* heures et *m* minutes.

- Q4. Écrivez un algorithme **LargeurImage** qui prend en entrée un tableau de tableaux de booléens codant une image en noir et blanc et qui renvoie la largeur de l'image correspondante. Par exemple, **LargeurImage**(*zigzag*) renvoie 4.
- Q4'. Même question pour l'algorithme **HauteurImage** (bien sûr **HauteurImage**(*zigzag*) renvoie 2).
- Q5. Écrivez un algorithme **CompteBlancsColonne** qui prend en entrée un tableau de booléens et renvoie le nombre de cases contenant VRAI dans ce tableau. Par exemple, **CompteBlancsColonne**({VRAI,FAUX,VRAI,VRAI}) renvoie 3.
- Q6. Sans faire de boucle (en utilisant un appel de l'algorithme **CompteBlancsColonne**), écrivez un algorithme **CompteNoirsColonne** qui prend en entrée un tableau de booléens et renvoie son nombre de cases contenant FAUX.
- Q7. Écrivez un algorithme **CompteBlancs** qui prend en entrée un tableau de tableaux de booléens codant une image et renvoie le nombre de pixels blancs dans l'image codée par ce tableau de tableau de booléens.
- Q8. On dispose d'un système qui permet d'afficher un pixel blanc pour 0.2€ par heure, et d'afficher un pixel noir pour 0.1€ par heure. Écrivez un algorithme **CoutAffichage** qui prend en entrée un tableau de tableaux de booléens *tabImage* codant une image et deux entiers *h* et *m*, et renvoie le coût d'affichage de l'image pendant *h* heures et *m* minutes.

- Q4. Écrivez un algorithme **LargeurImage** qui prend en entrée un tableau de tableaux de booléens codant une image en noir et blanc et qui renvoie la largeur de l'image correspondante. Par exemple, **LargeurImage**(*zigzag*) renvoie 4.
- Q4'. Même question pour l'algorithme **HauteurImage** (bien sûr **HauteurImage**(*zigzag*) renvoie 2).
- Q5. Écrivez un algorithme **CompteBlancsColonne** qui prend en entrée un tableau de booléens et renvoie le nombre de cases contenant VRAI dans ce tableau. Par exemple, **CompteBlancsColonne**({VRAI,FAUX,VRAI,VRAI}) renvoie 3.
- Q6. Sans faire de boucle (en utilisant un appel de l'algorithme **CompteBlancsColonne**), écrivez un algorithme **CompteNoirsColonne** qui prend en entrée un tableau de booléens et renvoie son nombre de cases contenant FAUX.
- Q7. Écrivez un algorithme **CompteBlancs** qui prend en entrée un tableau de tableaux de booléens codant une image et renvoie le nombre de pixels blancs dans l'image codée par ce tableau de tableau de booléens.
- Q8. On dispose d'un système qui permet d'afficher un pixel blanc pour 0.2€ par heure, et d'afficher un pixel noir pour 0.1€ par heure. Écrivez un algorithme **CoutAffichage** qui prend en entrée un tableau de tableaux de booléens *tabImage* codant une image et deux entiers *h* et *m*, et renvoie le coût d'affichage de l'image pendant *h* heures et *m* minutes.

- Q4. Écrivez un algorithme **LargeurImage** qui prend en entrée un tableau de tableaux de booléens codant une image en noir et blanc et qui renvoie la largeur de l'image correspondante. Par exemple, **LargeurImage**(*zigzag*) renvoie 4.
- Q4'. Même question pour l'algorithme **HauteurImage** (bien sûr **HauteurImage**(*zigzag*) renvoie 2).
- Q5. Écrivez un algorithme **CompteBlancsColonne** qui prend en entrée un tableau de booléens et renvoie le nombre de cases contenant VRAI dans ce tableau. Par exemple, **CompteBlancsColonne**({VRAI,FAUX,VRAI,VRAI}) renvoie 3.
- Q6. Sans faire de boucle (en utilisant un appel de l'algorithme **CompteBlancsColonne**), écrivez un algorithme **CompteNoirsColonne** qui prend en entrée un tableau de booléens et renvoie son nombre de cases contenant FAUX.
- Q7. Écrivez un algorithme **CompteBlancs** qui prend en entrée un tableau de tableaux de booléens codant une image et renvoie le nombre de pixels blancs dans l'image codée par ce tableau de tableau de booléens.
- Q8. On dispose d'un système qui permet d'afficher un pixel blanc pour 0.2€ par heure, et d'afficher un pixel noir pour 0.1€ par heure. Écrivez un algorithme **CoutAffichage** qui prend en entrée un tableau de tableaux de booléens *tabImage* codant une image et deux entiers *h* et *m*, et renvoie le coût d'affichage de l'image pendant *h* heures et *m* minutes.