LU3IN003 - PROJET
Un problème de tomographie discrète

Esther CHOI (3800370) et Vinh-Son PHO ()

November 15, 2020

Sommaire

1 Méthode incomplète de résolution

1.1 Première étape

Q1 Il suffit de regarder s'il existe $j \in \{1, ..., M-1\}$ tel que $T(j,k) =$ vrai. En effet cela signifierait qu'il existe un coloriage possible des $j+1$ premières cases avec la séquence complète $(s_1, ..., s_k)$.

Q2 Commençons par remarquer que $\forall j \in \{0, ..., M-1\}$ et $\forall l \in \{1, ..., k\}$, pour que $T(j,l)$ soit vrai, il faut que les $j+1$ premières cases contiennent au moins les l premiers blocs noirs en entier en plus des $l-1$ cases blanches pour séparer chaque bloc noir, c'est-à-dire qu'il faut :

$$j+1 \geq l-1+\sum_{i=1}^{l} s_i = l-1+s_l+\sum_{i=1}^{l-1} s_i \implies \boxed{j \geq l-1+s_l-1+\sum_{i=1}^{l-1} s_i}$$

- cas de base 1 : si $l=0$, cela signifie qu'il n'y a pas de blocs à placer. Donc il existe un coloriage possible pour les $j+$ premières cases : il suffit qu'elles soient toutes blanches.
- cas de base 2a : supposons $j < s_l - 1$
  - si $l=1$: alors $l-1+s_l-1+\sum_{i=1}^{l-1} s_i = s_1 - 1$
    D'après la remarque que l'on a faite, pour avoir $T(j,l) =$ vrai, il faudrait avoir $j \geq s_1 - 1$.
    Or, on a supposé $j < s_l - 1 = s_1 - 1$
    Donc pour $l=1$, $T(j,l) =$ faux
  - si $l \geq 2$: alors $l-1+s_l-1+\sum_{i=1}^{l-1} s_i > s_l - 1$ car $l-1>0$
    Pour avoir $T(j,l) =$ vrai, il faudrait avoir $j > s_l - 1$
    Or, on a supposé $j < s_l - 1$
    Donc pour $l \geq 2$, $T(j,l) =$ faux

  Conclusion : $\boxed{\forall l \geq 1, \text{ si } j < s_l - 1, \text{ alors } T(j,l) = \text{faux}}$
- cas de base 2b : supposons $j = s_l - 1$
  - si $l=1$: alors de même, $l-1+s_l-1+\sum_{i=1}^{l-1} s_i = s_1 - 1$
    D'après la remarque que l'on a faite, pour avoir $T(j,l) =$ vrai, il faudrait avoir $j \geq s_1 - 1$.
    Donc, en particulier pour $j = s_l - 1$, $T(j,l) =$ vrai
  - si $l \geq 2$: alors de même, $l-1+s_l-1+\sum_{i=1}^{l-1} s_i > s_l - 1$ car $l-1>0$
    Pour avoir $T(j,l) =$ vrai, il faudrait avoir $j > s_l - 1$

Or, on a supposé $j = s_l - 1$
Donc pour $l \geq 2$, $T(j,l) = \text{faux}$

Conclusion : $\boxed{\text{si } j = s_l - 1, \text{ alors } T(j,l) = \begin{cases} \text{vrai} & \text{si } l = 1 \\ \text{faux} & \text{si } l \geq 2 \end{cases}}$

Q3 On a la relation de récurrence suivante : $\boxed{T(j,l) = T(j-1,l) \vee T(j-s_l-1,l-1)}$
Avec comme cas de base : $\forall j \in \{0, ..., M-1\}, T(j,0) = \text{vrai}$, et $T(s_1 - 1, 1) = \text{vrai}$.
En effet :

- si on arrive à faire rentrer les l premiers blocs dans les j premières cases (c'est-à-dire si $T(j-1,l) = \text{vrai}$), alors on arrivera à les faire rentrer dans les $j+1$ premières cases (c'est-à-dire $T(j,l) = \text{vrai}$) en coloriant la j-ème case en blanc.

- si on arrive à faire rentrer les $l-1$ premiers blocs sur un certain nombre de cases j' (c'est-à-dire si $T(j', l-1) = \text{vrai}$), alors on pourra faire rentrer le bloc l si et seulement si $j \geq j' + s_l + 1$ (le $+1$ venant de la case blanche séparant le bloc $l-1$ du bloc l), donc en particulier si $j = j' + s_l + 1$, c'est-à-dire si $j' = j - s_l - 1$. Ainsi, on a $T(j,l) = \text{vrai}$, et la j-ème case est noire et correspond à la dernière case du bloc l.

- il suffit que l'une des deux conditions précédentes soit vraie pour que $T(j,l)$ soit égale à vrai, d'où le $\vee$.

- les cas de base sont les cas de base 1 et 2b pour $l = 1$ de la question précédente.

Q4 L'algorithme en pseudo-code est le suivant :

Algorithme 1 ColoriagePossibleRec

Entrée : $T, s = (s_1, ..., s_k), j, l$

Sortie : Retourne le tableau T tel que $T[j][l] = T(j, l)$. On suppose que pour le premier appel, $j = M - 1$ et $l = k$

1: Initialiser toutes les cases de T à -1
2: **Pour** j allant de 0 à $M - 1$ **faire**
3:    $T[j][0] \leftarrow$ vrai
4: **Fin pour**
5: $T[s_1 - 1][1] \leftarrow$ vrai
6: **Si** $j < s_{l-1} - 1$ **alors**
7:    $T[j][l] \leftarrow$ faux
8:    **Retourner** faux
9: **Sinon si** $j = s_{l-1} - 1$ **alors**
10:    **Si** $l = 1$ **alors**
11:       $T[j][l] \leftarrow$ vrai
12:       **Retourner** vrai
13:    **Sinon**
14:       $T[j][l] \leftarrow$ faux
15:       **Retourner** faux
16:    **Fin Si**
17: **Sinon**
18:    **Si** $(T[j-1][l] \neq -1)$ et $(T[j - s_{l-1} - 1][l-1] \neq -1)$ **alors**
19:       $T[j][l] \leftarrow (T[j-1][l]$ ou $T[j - s_{l-1} - 1][l-1])$
20:       **Retourner** $T[j][l]$
21:    **Sinon si** $T[j-1][l] \neq -1$ **alors**
22:       $T[j][l] \leftarrow ((T[j-1][l])$ ou $ColoriagePossibleRec(T, s, j - s_{l-1} - 1, l - 1))$
23:       **Retourner** $T[j][l]$
24:    **Sinon si** $T[j - s_{l-1} - 1][l-1] \neq -1$ **alors**
25:       $T[j][l] \leftarrow ((T[j - s_{l-1} - 1][l-1]$ ou $ColoriagePossibleRec(T, s, j - 1, l))$
26:       **Retourner** $T[j][l]$
27:    **Sinon**
28:       $T[j][l] \leftarrow (ColoriagePossibleRec(T, s, j - 1, l)$ ou $ColoriagePossibleRec(T, s, j - s_{l-1} - 1, l - 1))$
29:       **Retourner** $T[j][l]$
30:    **Fin Si**
31: **Fin Si**

- ligne 1 : initialisation de la matrice
- lignes 2-5 : cas de base du cas 2c
- lignes 6-8 : cas 1
- lignes 9-16 : cas 2a

- lignes 17-31 : cas 2c

En réalité, l'algorithme ne calcule pas nécessairement toutes les cases du tableau T : il ne calcule que les valeurs utiles, celles qui répondent à la question 1. En effet, si le résultat voulu est calculé, on le retourne ; sinon, on stocke progressivement les valeurs qui permettent le calculer. C'est le principe de la programmation dynamique.

1.2 Généralisation

Q5

Q6

Q7

1.3 Propagation

Q8

Q9

1.4 Tests

Q10

Q11