- Plus Grand Carré Blanc -

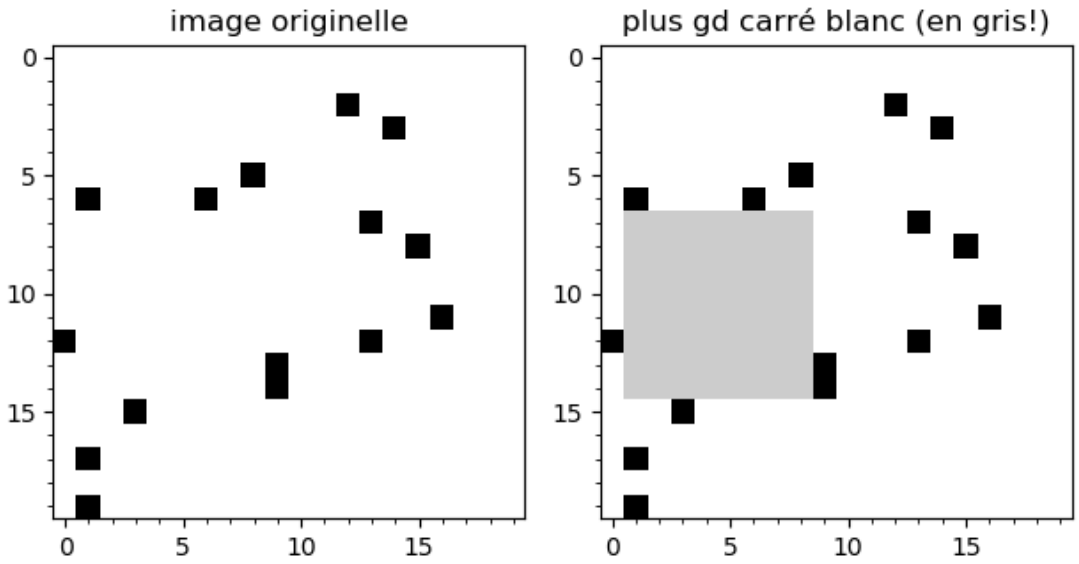
Le fichier *pgcb_ trame.py* commence avec le code ci-dessous qui génère une image dont certains pixels sont noirs, et les autres blancs.

```
import numpy as np
import matplotlib.pyplot as plt

def get_image(N,p=0.1):
    '''N : Taille de l'image
    p : proportion de points noirs, 0.1 par défaut'''
    image=np.random.random((N,N))>p
    return image.astype(float)


A=get_image(20)
```

Votre mission, si vous l'acceptez, est de trouver le plus grand carré blanc, c'est à dire la plus grande surface carrée ne contenant aucun point noir.

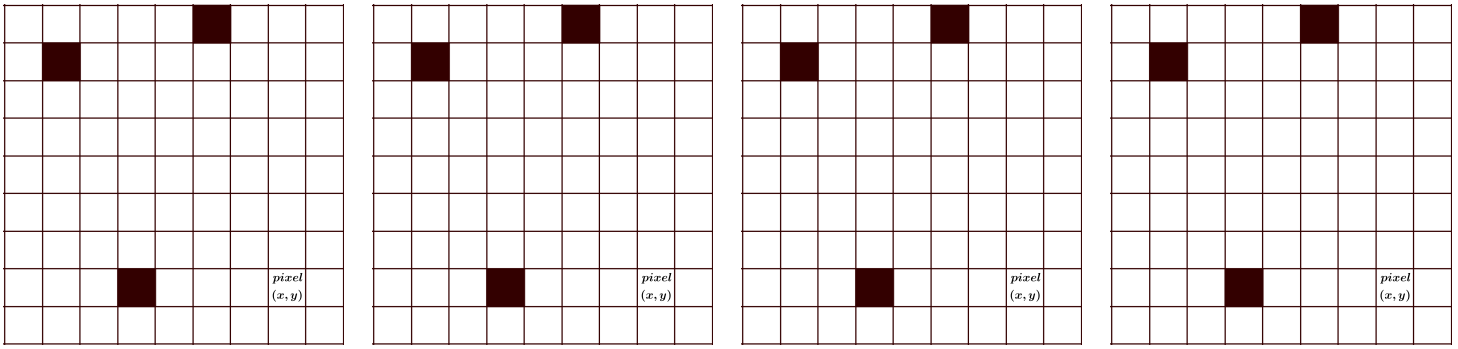


Pour cela, lorsqu'on considère un pixel de coordonnées (x, y), on note $T(x, y)$ la taille du plus grand carré blanc inclus dans l'image dont le coin en bas à droite correspond au pixel (x, y).

▼ Question 1

Déterminer les "cas de base", i.e. pour quels pixels la valeur de (x, y) la valeur $T(x, y)$ est elle simple à préciser.

▼ Question 2 Dans le cas général, exprimer $T(x, y)$ en fonction de $T(x-1, y-1), T(x-1, y)$ et $T(x, y-1)$.
On pourra s'aider en déterminant ces quatre quantités dans le quadrillage suivant :



▼ Script 3 Coder une fonction `pgcb_memo(A)` en programmation dynamique par mémoïsation qui renvoie le dictionnaire T contenant les valeurs $T(x, y)$ pour chaque pixel (x, y).

▼ Script 4 Coder une fonction `emplacement(A)` donnant en sortie une liste $[n, (x_0, y_0)]$ correspondant à la taille n "du" plus grand carré et aux coordonnées (x_0, y_0) du pixel lié à ce plus grand carré.

▼ Script 5 Utiliser la fonction `affiche(A)` présente dans le fichier *pgcb_ trame.py* pour visualiser votre résultat.

▼ Script 6 Coder une une fonction `pgcb_botomup(A)` ayant le même cahier des charges que `pgcb_memo(A)` mais avec une approche bottom up.