

```
import numpy as np
import matplotlib.pyplot as plt
import skimage.io
import maxflow.fastmin # used in question 12
```

```
### QUESTION 1
```

```
print(f'Question 1: {12*(5.12 + 7.16):0.3f}')
```

```
### QUESTION 3
```

```
L = np.array([28.9, 19.9, 13.7, 9.8, 7.0])
t = np.array([1, 2, 3, 4, 5])
print(f'Question 3: {np.sqrt(2*t[np.argmax(L*t)])}')
```

```
### QUESTION 4
```

```
d = np.loadtxt('../data/distances.txt')
lab = np.loadtxt('../data/labels.txt')
lab_c2 = lab[np.argmin(d, axis = 0)==1]
print(f'Question 4: {np.sum(lab_c2 == 1)/lab_c2.shape[0]:0.2f}')
```

```
### QUESTION 6 *
```

```
s = 1.7
t = np.array([[36, 13]]).T
theta = 140/180*np.pi
R = np.array([[np.cos(theta), -np.sin(theta)], [np.sin(theta), np.cos(theta)]])
```

```
p = np.loadtxt('../data/points_p.txt')
q = np.loadtxt('../data/points_q.txt')
p_ = R.T@(q-t)/s
```

```
# visualization
```

```
fig,ax = plt.subplots(1,2)
ax[0].plot(p[0], p[1], 'r.', q[0], q[1], 'b.')
ax[0].plot(np.stack((p[0], q[0])), np.stack((p[1], q[1])), 'k', linewidth = 0.3)
ax[0].set_aspect('equal')
ax[1].plot(p[0], p[1], 'r.', p_[0], p_[1], 'b.')
ax[1].plot(np.stack((p[0], p_[0])), np.stack((p[1], p_[1])), 'k', linewidth = 0.3)
ax[1].set_aspect('equal')
```

```
d = np.sqrt(np.sum((p - p_)**2,axis=0))
print(f'Question 6: {np.sum(d>2)}')
```

```
### QUESTION 9
```

```
print(f'Question 9: {np.sqrt((209-147)**2+(158-215)**2)}')
```

```
### QUESTION 10
```

```
print(f'Question 10: {(52-30)**2-(52-20)**2+(3-1)*125}')
```

```
### QUESTION 11 *
```

```
# Solution based on:
```

```
# https://lab.compute.dtu.dk/aia02506/weekly-
solutions/-/blob/master/Week05/week05_pycode/quiz_solutions.py
I = skimage.io.imread('../data/circly.png').astype(float)
mu = np.array([70, 120, 180], dtype=float)
beta = 100
```

```
U = (I.reshape(I.shape+(1,)) - mu.reshape(1,1,-1))**2
S0 = np.argmin(U, axis=2)
```

```
prior = beta * ((S0[1:,:]!=S0[:-1,:]).sum() + (S0[:,1:]!=S0[:, :-1]).sum())
likelihood = int(((mu[S0]-I)**2).sum())
```

```

fig, ax = plt.subplots(1,2)
ax[0].imshow(I, cmap=plt.cm.gray, vmin=0, vmax=255)
ax[1].imshow(S0, cmap=plt.cm.jet, vmin=0, vmax=2)
print(f'Question 11: {prior + likelihood}')

### QUESTION 12 *
# Solution based on:
# https://lab.compute.dtu.dk/aia02506/weekly-
solutions/-/blob/master/Week05/week05_pycode/dtu_binary.py
I = skimage.io.imread('../data/bony.png').astype(float)

mu = np.array([130, 190], dtype=float)
beta = 3000

# Graph with internal and external edges
U = (I.reshape(I.shape+(1,)) - mu.reshape(1,1,-1))**2
g = maxflow.Graph[float]()
nodeids = g.add_grid_nodes(I.shape)
g.add_grid_edges(nodeids, beta)
g.add_grid_tedges(nodeids, U[:, :, 1], U[:, :, 0])
g.maxflow()
S = g.get_grid_segments(nodeids)

# Visualization
fig, ax = plt.subplots()
ax.imshow(S)
print(f'Question 12: {S.sum()}')

### QUESTION 14 *
# Solution based on
# https://lab.compute.dtu.dk/aia02506/weekly-
solutions/-/blob/master/Week06/week06_pycode/quiz_solution.py
# and
# https://lab.compute.dtu.dk/aia02506/weekly-
solutions/-/blob/master/Week06/week06_pycode/plusplus_segmentation.py
I = skimage.io.imread('../data/frame.png').astype(float)/255
mask = np.zeros(I.shape, dtype=bool)
mask[I.shape[0]//2-40:I.shape[0]//2+40, I.shape[1]//2-40:I.shape[1]//2+40] = 1

m_in = np.mean(I[mask])
m_out = np.mean(I[~mask])

p = [I.shape[0]//2+39.5, I.shape[1]//2-40.5]
I_p = I[int(p[0]),int(p[1])]
f_ext = (m_in - m_out) * (2*I_p - m_in - m_out)

# Visualization
fig, ax = plt.subplots()
rgb = 0.5*(np.stack((I,I,I), axis=2) + np.stack((mask,mask,0*mask), axis=2))
ax.imshow(rgb)
ax.plot(p[1], p[0], 'co', markersize=10)
print(f'Question 14: {f_ext}')

### QUESTION 15
S = np.array([[0.1, 2.9], [1.2, 5.4], [3.3, 7.1], [3.5, 0.2], [1.4, 1.1]])
P = S[0] + 0.05*(S[1]+S[-1]-2*S[0]) + 0.1*(-S[2]-S[-2]+4*S[1]+4*S[-1]-6*S[0])
# Visualization
fig, ax = plt.subplots()
ax.plot(S[:,0], S[:,1], 'b-o', S[[0,-1],0], S[[0,-1],1], 'b:')
ax.plot(S[0,0], S[0,1], 'ro', P[0], P[1], 'co')
print(f'Question 15: {P}')

### QUESTION 17
I = np.loadtxt('../data/layers.txt')
bright = (20-I).sum(axis=1)

```

```
dark = I.sum(axis=1)
cost = [bright[0:s+1].sum() + dark[s+1:].sum() for s in range(I.shape[0])]
print(f'Question 17: {min(cost)}')
```

```
### QUESTION 18
```

```
yhat = np.array([0.5, 8.2, 6.9, -0.1, 0.3])
y = np.exp(yhat)
y /= y.sum()
print(f'Question 18: {-np.log(y[1])}')
```

```
### QUESTION 19
```

```
W1 = np.array([[0.2, -1.3], [-0.3, 1.8], [-1.7, 1.6]])
W2 = np.array([[-1.4, 1.5, -0.5, 0.9], [0.2, 1.2, -0.9, 1.7]])
hp = W1 @ np.array([1, 2.5])
yhat = W2 @ np.concatenate((np.array([1]), np.maximum(hp,0)))
y = np.exp(yhat)
y /= y.sum()
print(f'Question 19: {y[1]}')
```

```
### QUESTIION 20
```

```
print(f'Question 20: {(5*5+1)*8 + (200+1)*10}')
```