We provide sample questions for image segmentation and deep learning concepts.

1. Let us consider a graph with 4 nodes given by $V = \{a, b, c, d\}$, and edges $E = \{ab, bc, cd, da\}$. The edge costs are given as $\{w(a, b) = 2s, w(b, c) = s, w(c, d) = 2s, w(a, d) = s\}$, where $s$ is some positive integer. Find the best partition of the vertex set $V$ based on normalized cut cost.

2. We are given the following function:
   
   $$f(x_1, x_2, x_3) = -2x_1 + x_2 - 5x_3 - x_1x_3 - 4x_2x_3 - 5x_1x_2,$$
   
   where $x_1, x_2, \text{ and } x_3$ are Boolean variables. Our goal is to minimize the function using maxflow/mincut algorithm. Construct the network graph and show the costs for at least 2 different cuts.

3. Consider a linear threshold unit $T$ that is defined as follows. Let $x_1, x_2, \ldots, x_n$ be the real inputs to the linear threshold unit and let $w_1, w_2, \ldots w_n$ be the real weights and let $b$ be the bias terms. Then the output from $T$ will be 1 if $w_1x_1 + w_2x_2 \ldots w_nx_n + b \geq 0$ and 0 otherwise. Consider a function $f(x, y)$ that takes two real inputs $x \text{ and } y$ and gives a Boolean output in the following manner: $f(x, y) = 1 \text{ if } x \text{ and } y \text{ satisfy the constraint set } \{x \geq 0, x \leq 1, y \geq 0, y \leq 1\}$ or the set $\{x \geq 2, x \leq 3, y \geq 0, y \leq 1\}$. For other values of $x \text{ and } y, f(x, y) = 0$. Build a network to model $f(x, y)$ using linear threshold units. You are free to use as many linear threshold units as you need. Manually come up with weights and biases for each linear threshold unit.

4. We are given a deep neural network DNN for MNIST digit classification. The first layer is the input layer with 784 neurons coming from 28x28 image pixels. There is only one channel in the input. The second layer is a convolution layer with 3 filters, each of size $5 \times 5$ with stride 1. We do not use any padding for convolution. The output from the second layer passes through RELU activation unit ($\text{RELU}(z) = \max(z, 0)$). The third layer is a $2 \times 2$ max-pooling layer with no padding and stride 1. The fourth layer is a fully connected layer with 10 neurons for classifying the 10 digits. The final fifth layer is a softmax layer as shown below:

   $$\text{softmax}(z_1, \ldots, z_{10}) = \left(\frac{e^{z_1}}{\sum_{i=1}^{10} e^{z_i}}, \ldots, \frac{e^{z_{10}}}{\sum_{i=1}^{10} e^{z_i}}\right)$$

The softmax layer produces 10 outputs, where each output could vary from 0 to 1. After training, an input of digit 2 would produce an output vector ([0, 0, 1, 0, 0, 0, 0, 0, 0, 0]) for most of the time, depending on the accuracy that the network can produce. Show the dimensions...
\[ A = \{a, b, c, d\} \quad B = \{e, f\} \]

\[
NC(A/B) = \frac{cut_3(A/B)}{assoc(A, B)} + \frac{cut_3(A/B)}{assoc(B, A)}
\]

\[
= \frac{2S}{\sqrt{S}} + \frac{2S}{\sqrt{S}}
\]

\[ A = \{a, d\} \quad B = \{c\} \]

\[
NC(A/B) = \frac{cut_2(A/B)}{assoc(A, B)} + \frac{cut_2(A/B)}{assoc(B, A)}
\]

\[
= \frac{4S}{5A} + \frac{4S}{5A}
\]

\[ A = \{a, b, c, d\} \quad B = \{c\} \]

\[
NC(A/B) = \frac{8}{5} = 1.6
\]
\[ f(x_1, x_2, x_3) = -2x_1 + x_2 - 5x_3 - x_1x_3 - 4x_2x_3 - 5x_1x_2 \]

\[ = (-2x_1 + x_2 - 5x_3) - x_1x_3 + x_1 - x_1 \]
\[ - 4x_2x_3 + 4x_2 - 4x_2 \]
\[ - 5x_1x_2 + 5x_1 - 5x_1 \]

\[ = (\underbrace{-2x_1 + x_2 - 5x_3}_n) + (1 - x_3)x_1 - x_1 \]
\[ + 4(1 - x_3)x_2 - 4x_2 \]
\[ + 5(1 - x_2)x_1 - 5x_1 \]

\[ = -8x_1 - 3x_2 - 5x_3 + (1 - x_3)x_1 + 4(1 - x_3)x_2 + 5(1 - x_2)x_1 \]

\[ = \underbrace{8(1 - x_3)x_1 - 8}_n + \underbrace{3(1 - x_2)x_2 - 3}_v + \underbrace{5(1 - x_3)x_1 - 5}_s - 11(1 - x_3)x_1 + 4(1 - x_3)x_2 + 5(1 - x_2)x_1 \]

\[ \text{Constant} \rightarrow -16 \]
⇒ Costs for 2 different cuts.

Cut 1: \{ x_1, x_2, x_3 \} → Source, \{ x_3 \} → Sink
Cut cost = 8 + 3 + 5 = 16

Cut 2: \{ x_1 \} → Sink, \{ x_2, x_3 \} → Source
Cut cost = 1 + 5 + 3 + 5
= 14

⇒ Cost for 2 different cuts
without the constant (-16)

You are not asked to consider all possible cuts and find the minimum cut.
\[ \begin{align*}
\mathcal{G^3} & \quad \begin{cases} 
\{ x \geq 0, x \leq 1, y > 0, y \leq 1^3 
\end{cases} \\
\mathcal{G^3} & \quad \begin{cases} 
\{ x \geq 2, x \leq 3, y > 0, y \leq 1^3 
\end{cases}
\end{align*} \]
of the layers 2 and 3. What is the total number of parameters (weights and biases) in the network?

Once the DNN is trained for MNIST digit classification, manually add one more layer as the new output layer with just one neuron that classifies whether a given digit is a prime number or not. Manually assign the weights and biases for the last layer. [Hint: DNN is already trained to give the classification for the 10 digits. The additional last layer should just decide if the digit is a prime number or not based on the output from the previous layer.]

\[
\begin{align*}
(2^8 - 5) + 1 &= 24 \\
\text{# Parameters} &= 3 \times 5 \times 5 + 3 \times 23 \times 23 \times 10 \\
\text{Final layer} &= +3 \\
&\leq \text{if you include bias} + 10 \\
\text{weights} &= \text{for convolution} \\
\text{bias terms} &= \text{for prime numbers} \\
\text{bias term} &= 6 \\
0/1 \text{ is 1 when} \\
digit &= 2, 3, 5, 7
\end{align*}
\]