Question 1 (Model complexity for images.)
This question characterizes the model complexity of a simple model class for images. This
model class relies on the following assumptions:

- The image has $N \times N$ pixels. (That is, there are $N^2$ pixels, not $N$.)
- Pixel values are integer valued, and there are only $C$ allowable values. (That is, the alphabet is of size $C$.)
- The distribution of each pixel value relies only on the 4 pixels immediately above it, below it, to its left, and to its right. (To keep things simple, you can ignore pixels near the edges of an image, where some neighboring pixels extend beyond the image.)

(a) What is the model complexity for this model class?

(b) We reduce the size of the model class $C$ by requiring that models $c \in C$ satisfy a symmetry condition. One possible symmetry condition requires the value of the pixel to our left to be no larger than that to our right; this condition can be enforced by “flipping” the context if needed. What is the modified model complexity?

Question 2 (Clustering.)
This question deals with $k$-means clustering, which was used in Project 1.

(a) Explain in words how the $k$-means clustering algorithm works. You should not use code or mathematical equations.

(b) Does the output of $k$-means always contain exactly $k$ clusters?

(c) What is the sum-of-squares for $k$-means?

Question 3 (Nearest neighbors classification.)
This question considers a $k$-nearest neighbor classifier for binary classification. For a test point, the classifier assigns the majority class among the $k$ nearest neighbors based on Euclidean ($\ell_2$) distance. The figure below shows training data for two classes represented by plus and minus signs in $p = 2$ dimensions.
(a) What value of $k$ seems reasonable for this dataset? What is the resulting training error?

(b) Are there some values of $k$ that might be too large or too small for this dataset?

(c) Please sketch the 1-nearest neighbor decision boundary for this dataset.

**Question 4** (Trees.)

This question considers a binary search tree whose keys are stored in such a way that satisfies the following binary-search property:

Let $x$ be a node in a binary search tree. If $y$ is a node in the left subtree of $x$, then $y.key \leq x.key$. Else if $y$ is a node in the right subtree of $x$, then $y.key \geq x.key$.

(a) For the following set of keys, $\{1, 4, 5, 10, 16, 17, 21\}$, draw two possible binary search trees of two different heights. For each such tree, please indicate clearly which key is associated with each node of the tree.

(b) Below is an algorithm for a binary-search tree:

```
TREE-SEARCH(x, k)
1 if x == nil or k == x.key
2 return x
3 if k < x.key
4 return TREE-SEARCH(x.left, k)
5 else return TREE-SEARCH(x.right, k)
```

Suppose that we store integers in the range 1–1,000 in a binary-search tree, and want to search for the number 363. One of the following sequences cannot be the sequence of nodes examined. What is this impossible sequence? Remember to justify your answer.

(i) 924, 220, 911, 244, 898, 258, 362, 363.


(iii) 2, 399, 387, 219, 266, 382, 381, 278, 363.

(iv) 935, 278, 347, 621, 299, 392, 358, 363.

(c) What is the computational complexity for searching a binary-search tree that stores $N$ objects? Please respond to this question in terms of both $O(\cdot)$ and $\Omega(\cdot)$ notation.