1. learning algorithm

   a. We will use bit vectors for storing the item sets (x_i) and solution set (y). The bit vectors are size s, the total number of items

   b.
```
y <- empty set
For each item and class, x_i and t_i:
  If t_i equals +1: #is desk buyer
    y = union(y, x_i)
  Else: #not desk buyer
    y = difference(y, x_i) #remove any non-desk items from the solution set
Return y
```

   c. Since we are using bit vectors the set union and difference is O(1). Therefore, the algorithm is constant in terms of s and k and linear in terms of n.

2. decision surface

   a. $\mathbf{w}$ is the normal to the decision surface so a vector parallel to $\mathbf{w}$ is the vector between x_1 and x_2. So,

$$\begin{aligned}
\phi(x_1) &= [1, \sqrt{2}(-\sqrt{2}), (\sqrt{2})^2]^T \\
&= [1, -2, 2]^T \\
\phi(x_2) &= [1, \sqrt{2}(3\sqrt{2}), (3\sqrt{2})^2]^T \\
&= [1, 6, 18]^T \\
\mathbf{w}' &= \phi(x_2) - \phi(x_1) \\
&= [1, -2, 2]^T - [1, 6, 18]^T \\
\mathbf{w}' &= [0, 8, 16]
\end{aligned}$$

   b. Since $\mathbf{w}'$ is the vector between $\phi(x_1)$ and $\phi(x_2)$, the maximum margin will be the point halfway between these 2 points along $\mathbf{w}'$. So we are looking for $\frac{\|\mathbf{w}'\|}{2}$.

$$\begin{aligned}
\frac{\|\mathbf{w}'\|}{2} &= \frac{\sqrt{0^2 + 8^2 + 16^2}}{2} \\
&= \frac{\sqrt{320}}{2} \\
\frac{\|\mathbf{w}'\|}{2} &= 4\sqrt{5}
\end{aligned}$$

   c. (There are other ways to solve this) We know that $\frac{1}{\|\mathbf{w}\|} = 4\sqrt{5}$ and that $\mathbf{w}$ is parallel to so $\mathbf{w} = a\mathbf{w}'$ for some scalar a. This a is length of $\mathbf{w}'$ (to normalize it) times the distance to the decision boundary since we know that $\frac{1}{\|\mathbf{w}'\|} = \frac{1}{\|\mathbf{w}\|}$. They both equal 1.

$$\begin{aligned} a &= \frac{1}{8\sqrt{5}}\frac{1}{4\sqrt{5}} \\ &= \frac{1}{160} \end{aligned}$$

Then we plug this into our formula above:

$$\begin{aligned} \mathbf{w} &= a\mathbf{w}' \\ &= \frac{1}{160}[0, 8, 16]^T \\ \mathbf{w} &= [0, 1/20, 1/10]^T \end{aligned}$$

d. Now that we know $\mathbf{w}$, to find w_0 we just need to plug in $\mathbf{w}$, $\phi(x_1)$ (or $\phi(x_2)$), and t_1 (or t_2) into the formulas from the optimization formula:

$$\begin{aligned} t_1(\mathbf{w}^T\phi(x_1) + w_0) &= 1 \\ \mathbf{w}^T\phi(x_1) + w_0 &= \frac{1}{t_1} \\ w_0 &= \frac{1}{t_1} - \mathbf{w}^T\phi(x_1) \\ w_0 &= \frac{1}{-1} - [0, 1/20, 1/10]^T[1, -2, 2] \\ w_0 &= -1 - (0 - 1/10 + 2/10) \\ w_0 &= -1 - 1/10 \\ w_0 &= -\frac{11}{10} \end{aligned}$$

e. With $\mathbf{w}$ and w_0 we just need to expand $\phi(x)$ in our discriminant function:

$$\begin{aligned} f(x) &= w_0 + \mathbf{w}^T\phi(x) \\ &= -\frac{11}{10} + [0, 1/20, 1/10]^T[1, \sqrt{2}x, x^2] \\ &= \frac{1}{10}x^2 + \frac{\sqrt{2}}{20}x - \frac{11}{10} \end{aligned}$$

3. This was done in class

4.

a. The total volume of the space is 1. If we pick a point at random we have a 6% chance of being within the range.

b. Since we have 2 dimensions now. If we pick a point at random we have a

6% chance of hitting the range for *each* dimension. There are 2 dimensions. Thus, to hit both we need to randomly hit both ranges and have a $0.06 * 0.06 = 0.0036$ (0.36%) chance of hitting this range.

c. For 100 features, it is $0.06^100 = 6.533186x10^{-123}$

d. The issue is that if we make a prediction based on the proximity of the unknown point to one of the sample points, as the dimensionality increases, we are less and less likely to be near one of the points.

e. This can be solved by using a different distance measure. The calculations above are based on a Euclidean distance formula. You can use an alternative like cosine distance and if you do the calculations the probabilities won't scale so poorly