1a.

Let us initially consider n = 2 and k =2
There are two ways possible. If we consider the time units as 2.

For n =3 and k=2
We can have 4 ways possible to divide them into two clusters, hence the time taken will be 4.

If we go on considering like this we will have

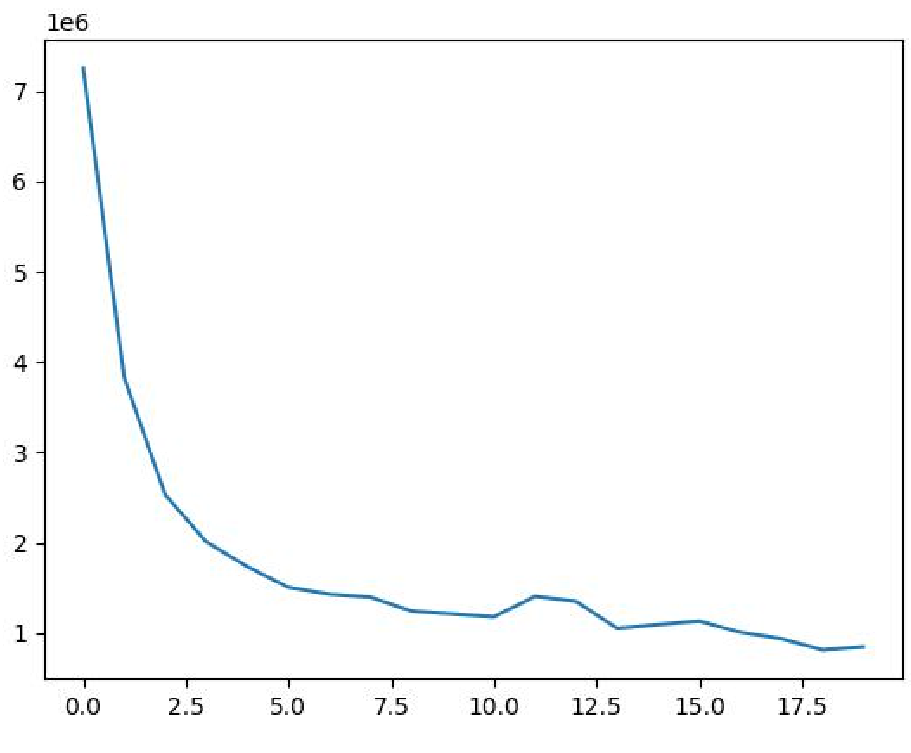
For n data points the time taken will be 2^{n-1} , which is exponential in nature.

Hence we can say that the Brute Force Algorithm is exponential in no of data points.

For the a,b,c,d parts of question 1 refer to the code submitted.

2a.

The plot for k vs k-means objective for different value of k is as follows



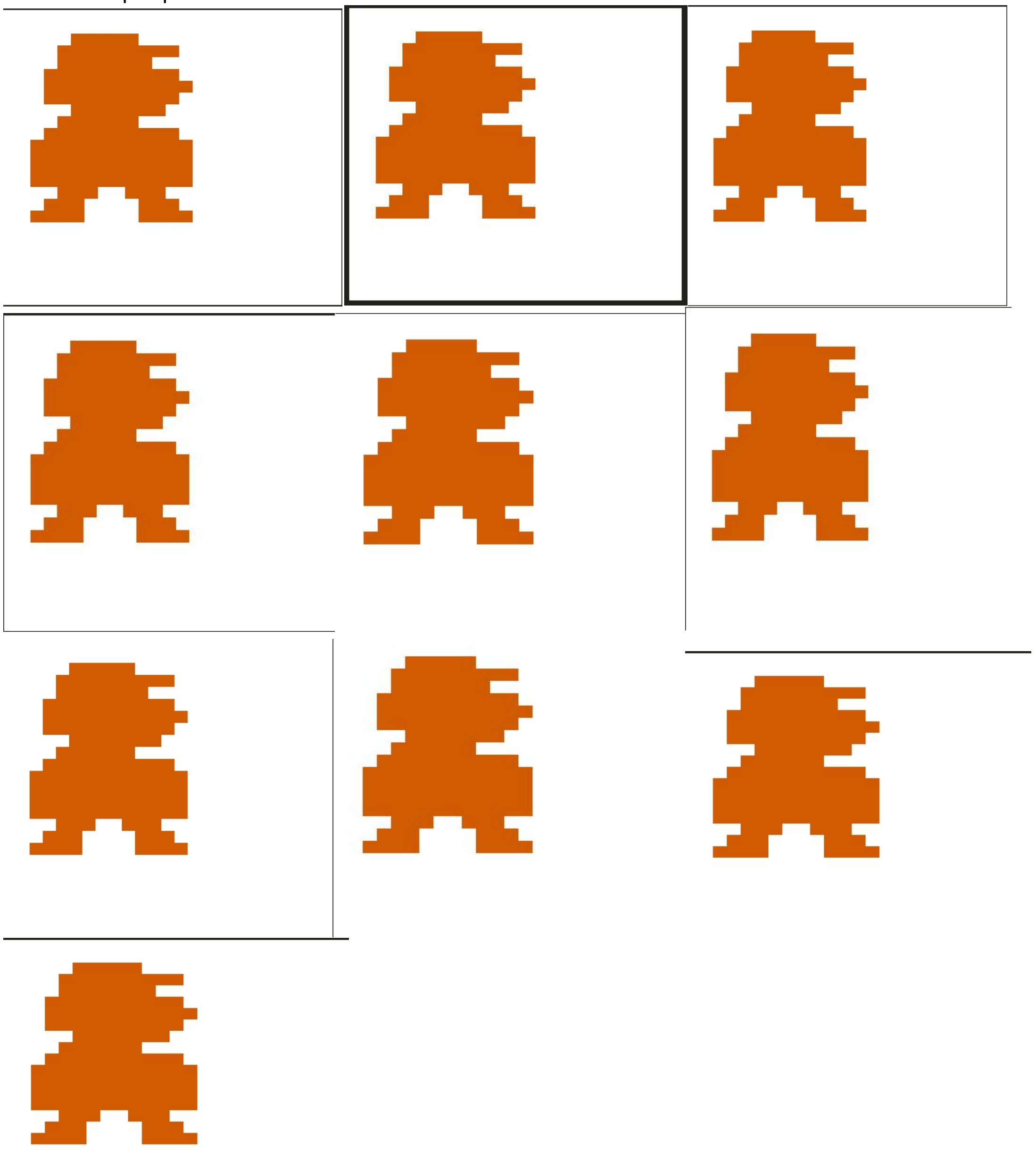
The y axis is the k-means objective and the x-axis is the k.

The code for the same is attached with the code. Look at the for_plots() function in code.

2b.

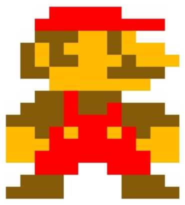
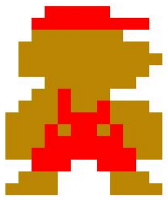
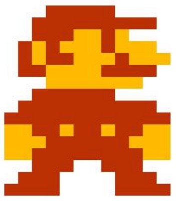
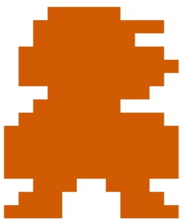
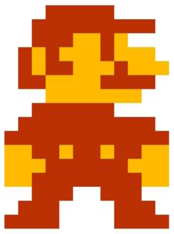
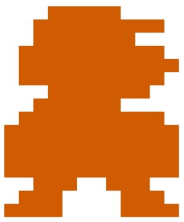
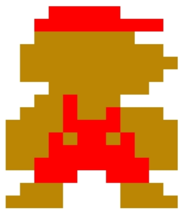
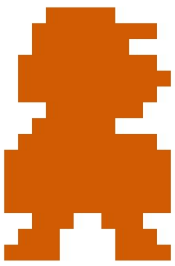
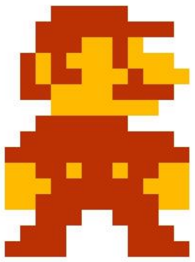
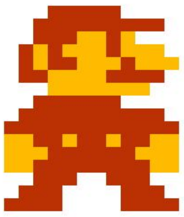
I have taken the test image 02 to get the plot for the k vs k-means objective. From the elbow or bend in the plot, I would choose k=2 as my value of k for this image. This value of k agrees with my intuition as well because if we have a look at the image I can see that there are mostly two colors which are dominant the white and blue and if we try to cluster the pixels based on these two colors we will get best results.

3a.
The ten output plots are as follows



The outputs didn't suffer from the problem of initialization. The no of iterations it took to converge varied but the centroids obtained at the end were almost similar. Since the no of clusters were low we are able to get similar results at the end of each iteration.

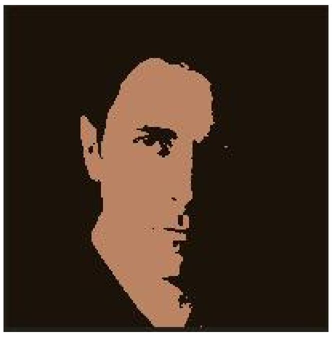
3b.

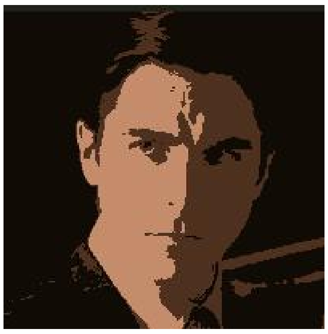
Some of the plots obtained suffered from the problem of bad initialization, they don't have more contrasting colors, they converged for a local optima. They were not able to be divided as 10 clusters because of the bad initialization. The problem with image 05 is that it has fewer than 10 differentiable colors and with bad initialization most of the pixels got assigned to less no of clusters and it converged.

3c.

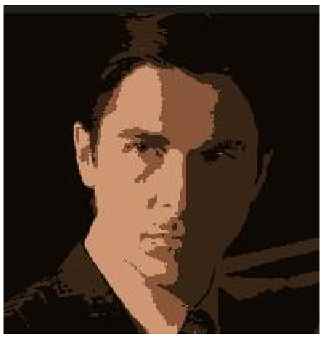
The 4 images obtained are
K=2



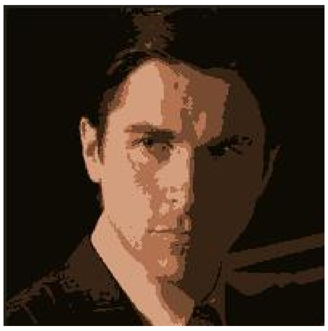
K=3



K=4



K=5



3d.

We can use the furthest point heuristic, k-means++ heuristic. Basically we will select the first center and for the second center we will take care that the second center is farther from the first center and so on. For example in this case if we take colour red as our first center we can take color blue as our second center and so on. In this way we can capture more contrasted images and cluster the pixels in a better way to get optimal outputs.