Assignment 3
Daniel Crawford
CS 6375 - Machine Learning

Average Log-likelihood on each dataset	No edges Bayesian Network	Bayesian Network using Chow-Liu	Mixture Tree using EM	Mixture Tree using RF
accidents	-303.09	-49.16	-49.16	**-48.28**
baudio	-198.69	-64.21	-64.21	**-62.94**
bnetflix	-122.44	-87.11	-87.11	**-86.38**
dna	-292.40	**-131.09**	**-131.09**	-132
jester	-125.49	-84.09	-84.09	**-82.48**
kdd	-478.69	-3.337	-3.337	**-3.297**
msnbc	-47.94	-9.471	-9.471	**-9.47**
nltcs	-24.89	-9.806	-9.806	**-9.67**
plants	-164.85	-24.23	-24.23	**-23.38**
r52	-5148.44	-146.24	-148.56	**-141.20**

Validation:
For both mixture tress, I validated k on the set [3,5,10]
For RF trees, I validated with values (assume test set size in n) [n / 3, n / 5, n / 10]

Choice of algorithm:
As we can see here, our random forest tree appears to have given us the best performance on each interval. It is likely that by creating k different trees we were able to reduce the variance in our algorithm, thus slightly improving our performance.
For our no edges Bayesian Network, we can see that we are getting very poor performance relative to the other algorithms for each test set. This is expected, because it is very underfit to the data, so it will not make good assumptions or generalizations.
Next we added a structure to our Bayesian Network with Chow-Liu. At this point, we see a large improvement in performance of our network relative to the no edges algorithm. This is due to the added structure being used.
Mixture trees with EM provided similar results to the Bayesian Network with Chow-Liu Trees, with the difference being very small (past two decimal points). These results are odd, and likely means there is something off in the algorithm restricting it from performing better.

In any case, it is safe to conclude that Mixture Trees using Random Forests were the most optimal algorithm for this problem given these results.

Reasoning of p_i assignment:
In my algorithm, I decided to randomly generate the probabilities of each of the trees instead of giving each tree equal weight. The reason for this is because randomization can show benefits in terms of model performance, as shown above. Even though there is no algorithm that is estimating the perfect weights for each network, we still see improved performance on each network.