Q3

a

In [7]:
```
import numpy as np
from sklearn import mixture
import math
```

In [38]:
```
def EM(xs, K, n):
    # initalization
    l = len(xs)
    mus = np.array([-2, 2])
    pis = np.array([.5, .5])
    sigmas = np.array([[1], [1]])
    r = np.zeros(l)
    for _ in range(n):
        # E-Step
        for i in range(l):
            if abs(xs[i] - mus[0]) >= abs(xs[i] - mus[1]):
                r[i] = 0
            else:
                r[i] = 1
        # M-Step
        mus[0] = np.mean(xs[r == 0])
        mus[1] = np.mean(xs[r == 1])
        sigmas[0] = 1/np.count_nonzero(r) * np.add(xs[r == 0], -mus[0]
        sigmas[1] = 1/(l-np.count_nonzero(r)) * np.add(xs[r == 1], - m
        pis = np.array([np.count_nonzero(r)/l, (l-np.count_nonzero(r))
        print("=========")
        print("mu = ", mus)
        print("sigma = ", sigmas)
        print("pi = ", pis)
        print("=========")
```

In [40]:
```

D = np.array([-9, -8, -7, 5, 7, 9])
EM(D, 2, 10)
```

```
=========
mu =  [ 7 -8]
sigma =  [[2]
 [0]]
```

```
pi =  [0.5 0.5]
=========
=========
mu =  [-8  7]
sigma =  [[0]
 [2]]
pi =  [0.5 0.5]
=========
=========
mu =  [ 7 -8]
sigma =  [[2]
 [0]]
pi =  [0.5 0.5]
=========
=========
mu =  [-8  7]
sigma =  [[0]
 [2]]
pi =  [0.5 0.5]
=========
=========
mu =  [ 7 -8]
sigma =  [[2]
 [0]]
pi =  [0.5 0.5]
=========
=========
mu =  [-8  7]
sigma =  [[0]
 [2]]
pi =  [0.5 0.5]
=========
=========
mu =  [ 7 -8]
sigma =  [[2]
 [0]]
pi =  [0.5 0.5]
=========
=========
mu =  [-8  7]
sigma =  [[0]
 [2]]
pi =  [0.5 0.5]
=========
=========
mu =  [ 7 -8]
sigma =  [[2]
 [0]]
pi =  [0.5 0.5]
=========
=========
mu =  [-8  7]
sigma =  [[0]
 [2]]
pi =  [0.5 0.5]
=========
```

b

For the given outputs above, after 1 iteration, we have p(x) = .5 * N(-8, 0) + .5 * N(7, 2)

c

we see that solution does not converge as it continues to oscillate

In [ ]: