In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

colors = ['red', 'blue', 'green', 'yellow', 'gray']
dataset = pd.read_csv('data-kmeans.csv')
data = dataset.values
x, y = data[:,0],data[:,1]
dataset
```

Out[1]:

	x	y
0	15	39
1	15	81
2	16	6
3	16	77
4	17	40
...	...	...
195	120	79
196	126	28
197	126	74
198	137	18
199	137	83

200 rows × 2 columns

In [2]:
```
def compute_distance(a, b):

    dist = np.power(np.sum(np.power(a - b,2),axis=1),0.5)

    return dist
```

In [3]:
```
def compute_centroid(Z, cluster_num):
    centroids = []
    for cluster in range(cluster_num) :
        centroids.append(np.mean(Z[Z[:,2]==cluster][:,:2], axis=0))
    return np.array(centroids)
```

In [4]:
```
def compute_label(z, M):
    dists = []
    for m in M :
        dists.append(compute_distance(z,m))
    dists = np.array(dists)
    dists = np.reshape(dists,[len(M),len(z)],order='F').T
    label = np.argmin(dists, axis=1)
    return label
```

In [5]:
```
def compute_loss(clusters, centroids) :
    loss = 0
    for i in range(len(centroids)) :
        cluster = clusters[clusters[:,2]==i][:,:2]
        loss += np.sum(compute_distance(cluster, centroids[i]))
    loss /= len(clusters)
    return loss
```

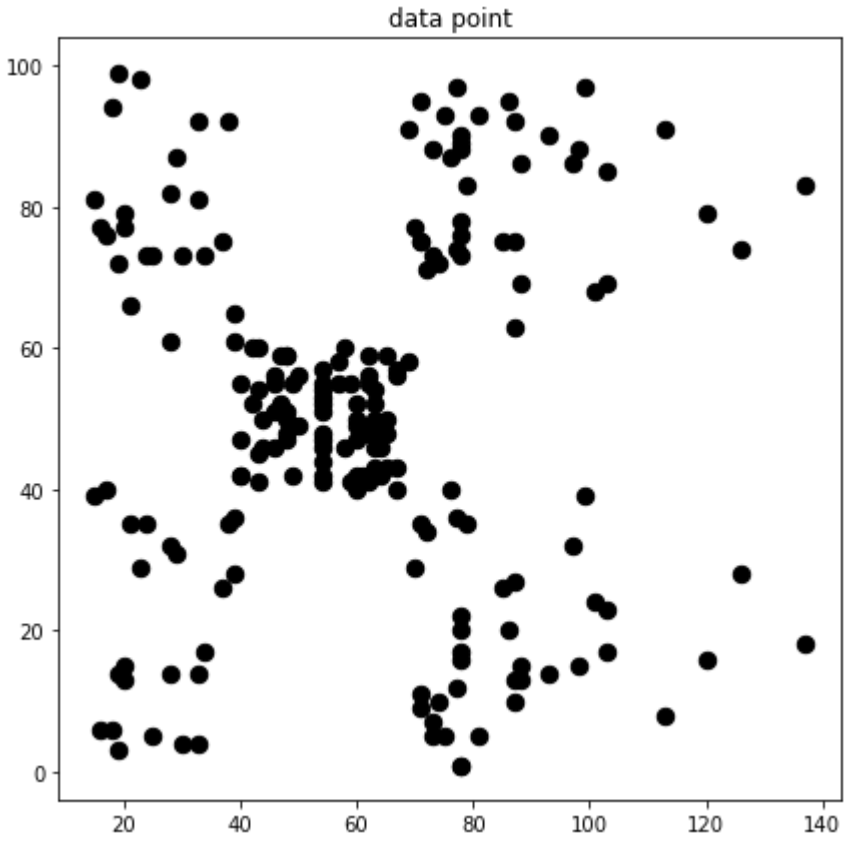
In [6]:
```
# init clusters
n = len(data)
clusters = np.append(data,np.ones(n)[:,None], axis=1)
cluster_num = 5
for idx in range(n) :
    rand_num = np.random.randint(cluster_num)
    clusters[idx][2] = rand_num
centroids = compute_centroid(clusters, cluster_num)
inital_clusters = np.copy(clusters)
inital_centroids = np.copy(centroids)
centroids
```

Out[6]:
```
array([[59.05714286, 55.71428571],
       [54.1       , 43.7       ],
       [61.14583333, 50.64583333],
       [66.91836735, 51.81632653],
       [59.53571429, 49.        ]])
```

In [7]:
```
n = 20
L_iters = [-1 for _ in range(n)]
cent_dist = [compute_distance(centroids,[0,0])]
for idx in range(n) :
    clusters[:,2] = compute_label(clusters[:,:2], centroids)
    centroids = compute_centroid(clusters, cluster_num)
    L_iters[idx] = compute_loss(clusters, centroids)
    cent_dist.append(compute_distance(centroids,[0,0]))
cent_dist = np.array(cent_dist)
```

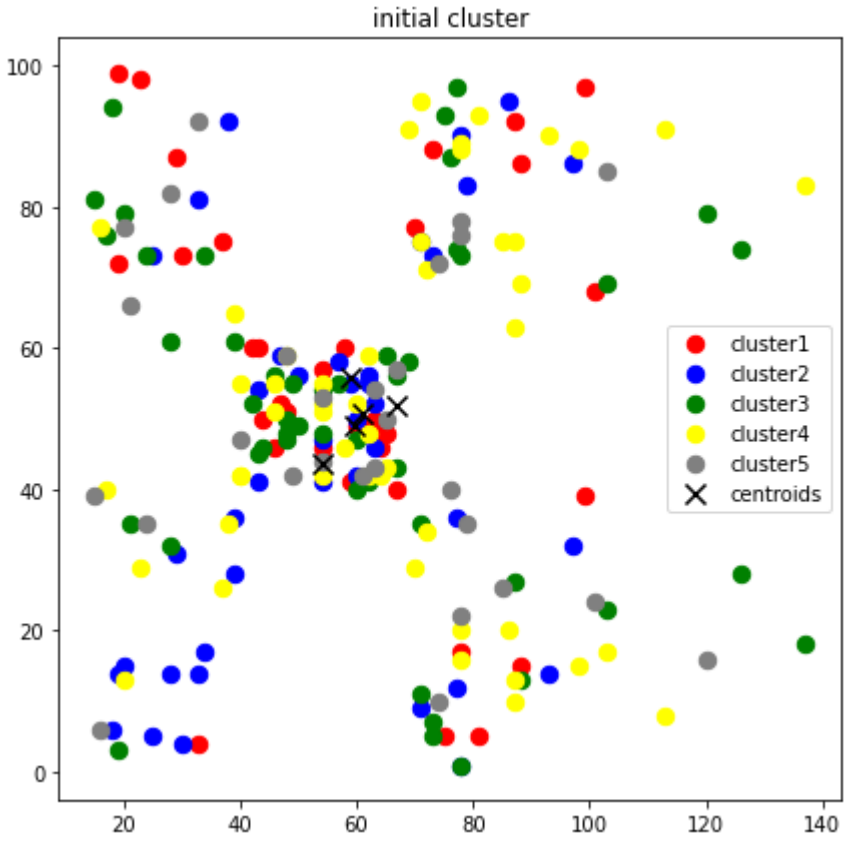
In [8]:
```
plt.figure(figsize=(7,7))
plt.title('data point')
plt.scatter(x,y, c='black', s=70)
```

Out[8]: <matplotlib.collections.PathCollection at 0x24f298b5430>



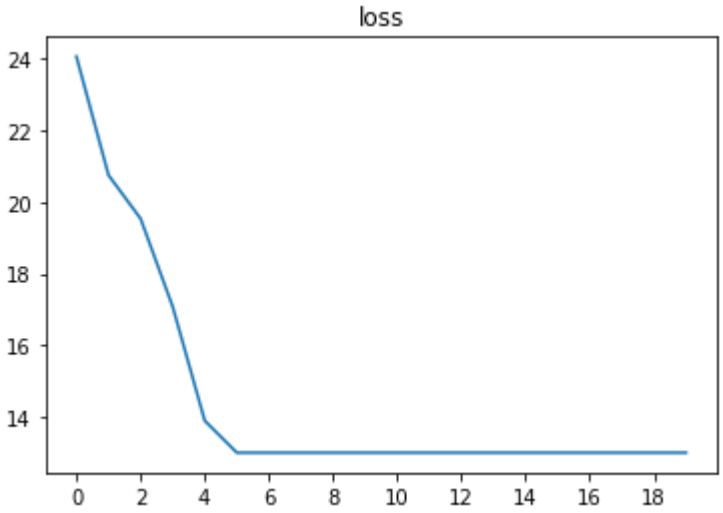
In [9]:
```
plt.figure(figsize=(7,7))
plt.title('initial cluster')
legends = ['cluster{}'.format(idx+1) for idx in range(cluster_num)]
legends.append('centroids')
for idx in range(cluster_num) :
    inital_cluster = inital_clusters[inital_clusters[:,2]==idx]
    plt.scatter(inital_cluster[:,0],inital_cluster[:,1], s=70, c=colors[idx])
plt.scatter(inital_centroids[:,0], inital_centroids[:,1], marker='x', c='black', s=100
plt.legend(legends)
```

Out[9]: <matplotlib.legend.Legend at 0x24f29970fa0>



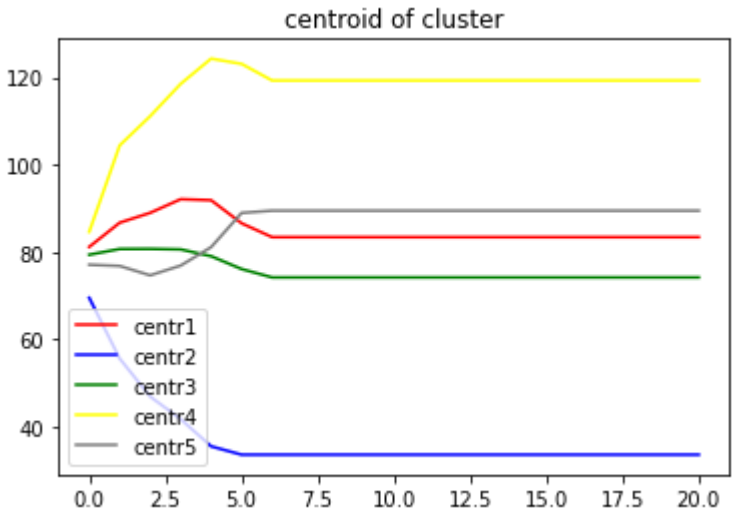
In [10]:
```
plt.title('loss')
plt.xticks(np.arange(0,len(L_iters),2))
plt.plot(L_iters)
```

Out[10]: [<matplotlib.lines.Line2D at 0x24f299ec0d0>]



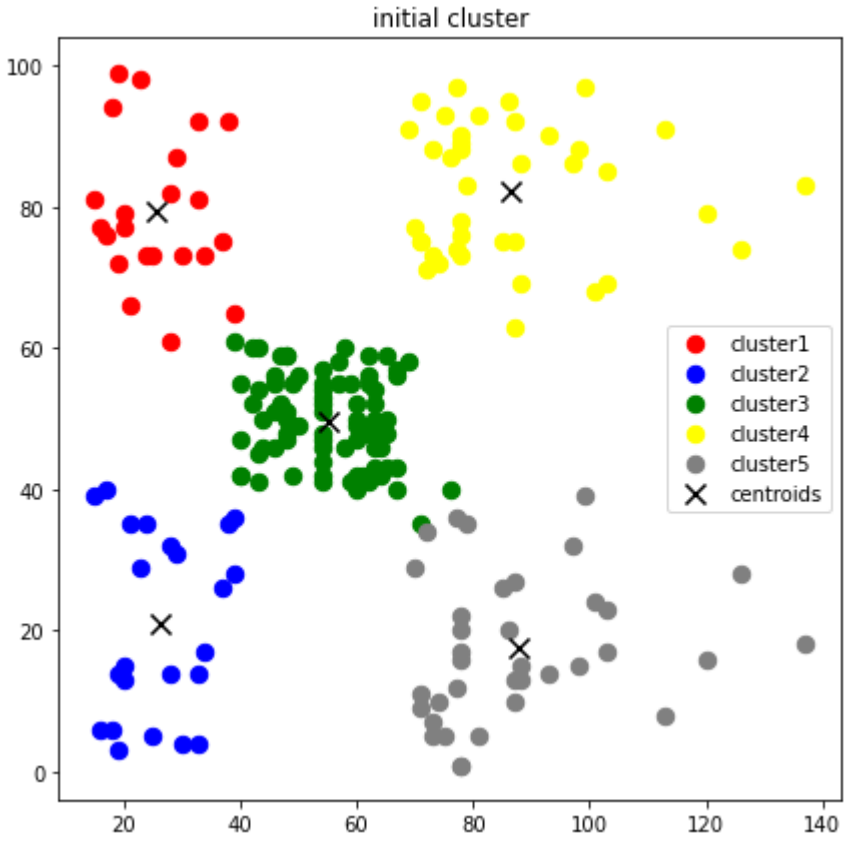
In [11]:
```
plt.title('centroid of cluster')
legends = ['centr{}'.format(idx+1) for idx in range(cluster_num)]
for idx in range(cluster_num) :
    plt.plot(cent_dist[:,idx], color=colors[idx])
plt.legend(legends)
```

Out[11]: <matplotlib.legend.Legend at 0x24f29a43520>



In [12]:
```
plt.figure(figsize=(7,7))
plt.title('initial cluster')
legends = ['cluster{}'.format(idx+1) for idx in range(cluster_num)]
legends.append('centroids')
for idx in range(cluster_num) :
    cluster = clusters[clusters[:,2]==idx]
    plt.scatter(cluster[:,0],cluster[:,1], s=70, c=colors[idx])
plt.scatter(centroids[:,0], centroids[:,1], marker='x', c='black', s=100)
plt.legend(legends)
```

Out[12]: <matplotlib.legend.Legend at 0x24f29a94bb0>



In [ ]:
```
```