In [ ]:

```
https://nlp.stanford.edu/IR-book/newslides.html (KNN, Rocchio)
```

KNN

In [53]:

```
from random import randint

N = 100
D = list((randint(1,100),randint(1,100)) for _ in range(N))
C = list(0 if sum(d) > 100 else 1 for d in D)
sample = (50, 50)
```

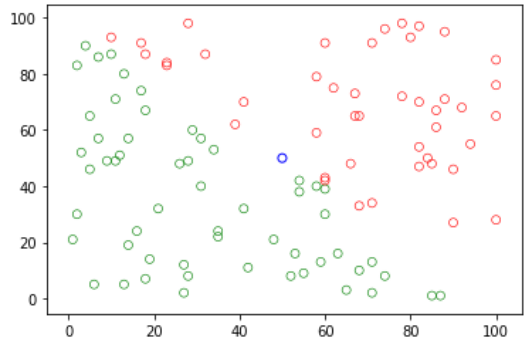
In [58]:

```
import matplotlib.pyplot as plt

X = [d[0] for d in D]
Y = [d[1] for d in D]
similarity = [0.0 for _ in range(N)]
colorMap = ["r","g","b"]

#Class 0 => r ...
for i in range(N):
    plt.scatter(X[i],Y[i],facecolor="none", edgecolors=colorMap[C[i]], linewidths=.6)
    #similarity[i] = euc(D[i], sample) #유사도 계산
    similarity[i] = cos(D[i], sample)

plt.scatter(sample[0], sample[1], facecolor="none", edgecolors=colorMap[-1]) #target : 퍼렁이
print(plt.show())
```



None

In [52]:

```
from math import sqrt

#relevance(d,q) => sim(d,q), in vs
#유클리드 거리
#Not numpy

def euc(x1,x2):
    return sqrt((x1[0] - x2[0])**2 + (x1[1] - x2[1])**2)

def cos(x1,x2):
    innerProduct = x1[0]*x2[0] + x1[1]*x2[1]
    return innerProduct/(euc(x1, (0,0))*euc(x2, (0,0)))
```

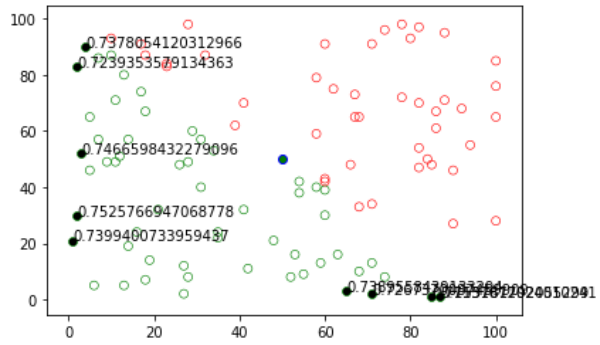
In [59]:

```
K = 9 #sqrt(N) #C = 짝수, K = 홀수 <- 보편적 상황에서
candidates = sorted(list(enumerate(similarity)), key=lambda x:x[1])[:K] #인덱스
#[(i, s) for i,s in zip(range(N), similarity)]
candidateKeys = [c[0] for c in candidates]
candidateClass = list(0 for _ in range(2))

for i in candidateKeys:
    candidateClass[C[i]] += 1

for i in range(N):
    plt.scatter(X[i],Y[i],facecolor="k" if i in candidateKeys else "none", W
                edgecolors=colorMap[C[i]], linewidths=.6)
    #similarity[i] = euc(D[i], sample) #유사도 계산
    #similarity[i] = cos(D[i], sample)
    if i in candidateKeys:
        plt.text(X[i], Y[i], candidates[candidateKeys.index(i)][1])

plt.scatter(sample[0], sample[1], facecolor="r" if candidateClass[0] > candidateClass[1] else
"g", W
            edgecolors=colorMap[-1]) #target : 퍼렁이
print(plt.show())
# for i in range(0,100):
#     print("%d번째" %i, similarity[i])
```



None