Problem 3

a)

$$\begin{aligned} c(s[0:i],t[0:j]) = min(&c(s[0:i-1],t[0:j])+2,\\ &c(s[0:i],t[0:j-1])+2,\\ &c(s[0:i-1],t[0:j-1]))+(1-\mathbb{1}_{s[i]=t[j]}) \end{aligned}$$

b)Initialized a cost matrix F of size $|s|+1 \times |t|+1$:
set $F[0,0]=0$, $F[i,0]=F[i-1,0]+2$for $i=1...|s|$, $F[0,j]=F[0,j-1]+2$for $j=1...|t|$

```
$for\ i=1...|s|$
    $forj=1...|t|$
      $F(i,j)=min(F[i-1,j]+2,F[i,j-1]+2,F[i-1,j-1]+(1-\mathbb{1}_{s[i]=t[j]})$
```

return $F[|s|,|t|]$

The time complexity of this algorithm is $\mathcal{O}(|s||t|)$, or in terms of the detailed steps $|s|+|t|+1+(3+3log(3))*|s||t|$

Problem 4

In [1]:

```
import numpy as np
import sys
from tqdm import tnrange

file = open('dna.txt','r')
s = file.readline()
s = s[:-1]
t = file.readline()
file.close()
```

In [ ]:

```
def cost(s,t):
    F = np.zeros((len(s)+1,len(t)+1))
    for i in range(1,len(s)+1):
        F[i,0] = F[i-1,0] + 2

    for j in range(1,len(t)+1):
        F[0,j] = F[0,j-1] + 2

    for i in range(1,len(s)+1):
        for j in range(1,len(t)+1):
            F[i,j] = min(F[i-1,j]+2,F[i,j-1]+2,F[i-1,j-1]+(1-int(s[i-1]==t[j-1])))
    return F[-1,-1]
```

In [ ]:

```
c_s_t = cost(s,t)
```

In [4]:

```
c_s_t/len(s)
```

```
Out[4]: 0.0223
```

These two sequences are serving the same function since they have less than 2.23% difference