In [51]:

```
def SymbolToNumber(symbol):
    StN = {"A":0,"C":1,"G":2,"T":3}
    return StN[symbol]

def PatternToNumber(Pattern):
    if not Pattern:
        return 0
    symbol = Pattern[-1]
    prefix = Pattern[0:-1]
    return ((4*PatternToNumber(prefix))+SymbolToNumber(symbol))

print(PatternToNumber("CTTCTCACGTACAACAAAATC"))
```

```
2161555804173
```

In [52]:

```
def NumberToSymbol(number):
    StN = {0:"A",1:"C",2:"G",3:"T"}
    return StN[number]

def NumberToPattern(index,k):
    if k == 1:
        return NumberToSymbol(index)
    prefixIndex = int(index/4)
    r = index%4
    symbol = NumberToSymbol(r)
    PrefixPattern = NumberToPattern(prefixIndex,k-1)
    return PrefixPattern + symbol

print(NumberToPattern(5353,7))
```

```
CCATGGC
```

In [53]:

```
def SymbolToNumber(symbol):
    StN = {"A":0,"C":1,"G":2,"T":3}
    return StN[symbol]

def PatternToNumber(Pattern):
    if not Pattern:
        return 0
    symbol = Pattern[-1]
    prefix = Pattern[0:-1]
    return ((4*PatternToNumber(prefix))+SymbolToNumber(symbol))

def ComputingFrequencies(text,k):
    FrequencyArray =[]
    for i in range(0,((4**k))):
        FrequencyArray.append(0)
    for i in range(0,(len(text)-1)):
        pattern = text[i:(i+k)]
        j = PatternToNumber(pattern)
        FrequencyArray[j] = FrequencyArray[j]+1
    return FrequencyArray

FrequencyArray = ComputingFrequencies("ACGCGGCTCTGAAA",2)
print(FrequencyArray)
```

```
[2, 1, 0, 0, 0, 0, 2, 2, 1, 2, 1, 0, 0, 1, 1, 0]
```

In [54]:

```
def SymbolToNumber(symbol):
    StN = {"A":0,"C":1,"G":2,"T":3}
    return StN[symbol]

def PatternToNumber(Pattern):
    if not Pattern:
        return 0
    symbol = Pattern[-1]
    prefix = Pattern[0:-1]
    return ((4*PatternToNumber(prefix))+SymbolToNumber(symbol))

def NumberToSymbol(number):
    StN = {0:"A",1:"C",2:"G",3:"T"}
    return StN[number]

def NumberToPattern(index,k):
    if k == 1:
        return NumberToSymbol(index)
    prefixIndex = int(index/4)
    r = index%4
    symbol = NumberToSymbol(r)
    PrefixPattern = NumberToPattern(prefixIndex,k-1)
    return PrefixPattern + symbol

def ComputingFrequencies(text,k):
    FrequencyArray =[]
    for i in range(0,((4**k))):
        FrequencyArray.append(0)
    for i in range(0,(len(text)-1)):
        pattern = text[i:(i+k)]
        j = PatternToNumber(pattern)
        FrequencyArray[j] = FrequencyArray[j]+1
    return FrequencyArray

def FasterFrequentWords(text,k):
    FrequentPatterns = []
    FrequencyArray = ComputingFrequencies(text,k)
    maxCount = max(FrequencyArray)
    for i in range(0,(4**k)):
        if FrequencyArray[i] == maxCount:
             pattern = NumberToPattern(i,k)
             FrequentPatterns.append(pattern)
    return FrequentPatterns

print(FasterFrequentWords("ACGCGGCTCTGAAA",2))
```

```
['AA', 'CG', 'CT', 'GC']
```