HOMEWORK 3 – Q1

MINGLANG XIE

z5228006

1. After the success of your latest research project in mythical DNA, you have gained the attention of a most diabolical creature: Medusa. Medusa has snakes instead of hair. Each of her snakes' DNA is represented by an upper-case string of letters. Each letter is one of S, N, A, K or E. Your extensive research shows that a snake's venom level depends on its DNA. A snake has venom level x if its DNA:

- has exactly 5x letters
- begins with x copies of the letter S
- then has x copies of the letter N
- then has x copies of the letter A
- then has x copies of the letter K
- ends with x copies of the letter E.

For example, a snake with venom level 1 has DNA SNAKE, while a snake that has venom level 3 has DNA SSSNNNAAAKKKEEE. If a snake's DNA does not fit the format described above, it has a venom level of 0. Medusa would like your help making her snakes venomous, by deleting zero or more letters from their DNA. Given a snake's DNA, can you work out the maximum venom level this snake could have? Your algorithm should run in time $O(nlog(n))$.

Solution:
Count the numbers n_S, n_N, n_A, n_K, n_E of occurrences of each letter S, N, A, K, E in the original sequence and let $M = \min\{n_S, n_N, n_A, n_K, n_E\}$. Clearly, the largest possible venom level L satisfies $L \le M$.
Let's begin with $L = M$, we iterate the given DNA (search for S, N, A, K, E in order), and count the number of S at the beginning of the DNA, when the count equal to L, change the count to be zero, then search for N, when the count equal to L, etc. If succeed to find out a DNA that has venom level L, L is the maximum venom level of this snake could have. If not, we then try to see if $L = \frac{M}{2}$ works. If it does, we see if $L = \frac{M}{2} + \frac{\frac{M}{2}}{2} = \frac{3M}{4}$ also works, etc.

This strategy is clearly optimal, as we always make decision on the level, and find that the DNA match for this level, if it's matching then make the level higher.

```
match(L):
    count = 0
    list = [S, N, A, K, E]
    list_index = 0
    for i in DNA:
        if i == list[list_index]:
            count += 1
        if count == L:
            list_index += 1
    if list_index == len(list):
        return True
   return false

searchMaxLevel(M, L, S):
    if M==L and match(L):
        return L
   if !match():
        M = L
        searchMaxLevel(M,$S+\frac{L-S}{2}$, S)
    else:
        S = L
        searchMaxLevel(M, $L+\frac{M-L}{2}$, S)

main():
  n = []                  // $n_S, n_N, n_A, n_K, n_E$
  list = [S, N, A, K, E]
  for i in DNA:
    n[list.index(i)] += 1
  M = min(n)

  searchMaxLevel(M,M,0)
```