Matthew Soicher
z5160737

Question 1

Our aim is that given a DNA string, we need to maximise its venom level by deleting zero or more letters.

First, we sum each number of letters, n_s, n_n, n_a, n_k, n_e in our string.

Since our DNA string must contain the **same quantity** of each letter for it to achieve non-zero venom level, and since we cannot add any letters, the maximum amount of each letter we can have is given by the smallest $n_s \dots n_e$ as shown above.

And so, the maximum venom level L we can achieve is M: $min\{n_s, n_n, n_a, n_k, n_e\}$

Now our plan is to use a **greedy strategy** to delete such letters in our original string such that the amount of each letter, $n_s \dots n_e = M$ to maximise L.

The problem is that our string may not be in the order that is required.

So, the strategy is that we perform a **binary search** to check if our string fits the requirements, initially for $L = M$. (M refers to the maximum quantity of each letter to achieve a non-zero venom level)

If it does, we return L, but if not, we continue and check for $L = M/2$. If the requirement fits for $M/2$, we check for $L = 3M/4$, and continue halving and checking the respective side as described in order to find the largest L.

And so, we run in $O(n)$ time for the search through the string to find the quantity of each $n_s \dots n_e$ and $O(logn)$ for the binary search, resulting in a $O(nlogn)$ algorithm