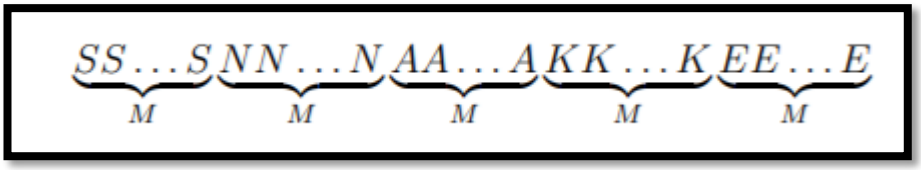
Dheeraj Satya Sushant Viswanadham
(z5204820)

COMP3121 Assignment 3 – Question 1

1) Our task here is to find a way to make the snakes more venomous by deleting zero or more letters from the given DNA of each snake, and our algorithm should run in $O(nlog(n))$ time.

See given diagram as an overall visual example: (note this diagram is on the hints sheet).



One approach to solve this problem is to count the number of occurrences of each letter (i.e. n_s, n_n, n_a, n_k, n_e) that occurs in the given string. Once we have that, we would get our set M which is the minimum number of times that each letter S, N, A, K and E occurs i.e. $M = \{n_s, n_n, n_a, n_k, n_e\}$. This would clearly be the largest possible venom level of L which would satisfy $L \leq N$, where N is our count of the total letters in the string. From here we can observe that if any one of our minimum occurrences of the letters is 0 then our return output will automatically be 0 as we are missing a letter that is required, meaning no matter how many letters we delete, the venom level will always be 0. Traversing through the DNA string would be $\mathrm{O(n)}$ time.

On the other hand, if we have at least n = 1 occurrences for each letter in the DNA string, we also have to account for whether these occurrences are in order i.e. 'S…N…A…K…E' otherwise if they were instead 'N…K…E…S…A' then there would obviously be a venom level of 0.

Hence, our starting point needs to be at the first occurrence of the letter S and we can disregard any occurrences of the other letters before this. Once we have established this, we can then employ a greedy strategy to see where our first n occurrences of S stop happening and if so, where our next letter N occurs, and so on. For example as below (where our underline is what we are ignoring/removing) is what we are aiming for. We will end the algorithm when we reach the last occurrence of E in the entire DNA string.

NNNAEEESSKKKKEENNNKKEEANNNKKSSSEEKKKSSS

(and this could be repeated over and over in the DNA string)

We can employ a recursive alogorithm for this and once we reach the last occurrence of E in the entire DNA string, we can find the largest number of occurrences of the letters within this. By using the above strategy to see if it is possible to delete some letters in between to increase our x (which is the copies of each letter occurring in order), we can observe if our L = M. If so, we have succeeded in finding the largest possible level of venom otherwise we can go to the next best stage and use a binary algorithm where we are observing if we can see L = M/2 (as we are splitting the maximum into two). If that works then we can then see if L = (¾)*M also works (where we scan the next upper half i.e. ¼ of M), and so on until we find the largest possible M. This binary search algorithm would take $O(\log(n))$ time.

Hence, our overall time complexity would be $\boldsymbol{O(\mathbf{nlog(n)})}$ as we are traversing through the DNA string and checking if we can increase our M at the same time using binary search.

End of Solution