Q1

Denote sequence A as a snake's DNA. First we find from venom level n ($n >= 1$), and denote level N DNA as sequence B, which means sequence B correspond to 'SNAKE' in level 1, 'SSNNAAKKEE' in level 2, 'S*nN*nA*nK*nE*n' in level n.

Then, we use hash map(denote $hash\_set$)to traverse sequence A and store indexes of each different word (i.e. keys are the different words and values are lists include indexes when this word occur in sequence A), this step will run in $O(n)$.

Next, we match words in sequence B in order. $B[i]$ must be behind $B[i-1]$, so when we do matching, we should find an index in $hash\_set[s[i]]$ which is bigger than index of $s[i-1]$ that you have found before.So we use binary search on $hash\_set[s[i]]$ that matches the left boundary (that is, match the first number greater than target value) to find the first value in the current letter index list greater than the previous letter index. This step will run in time $O(nlogn)$.

If all words in sequence B are match successfully, which means we can deleting zero or more letters from their DNA and make this snake venomous, and set the variable maximum_venom_level to n.

We do this by setting n equal to 1 and repeat this, when match successfully, by setting sequence B to level $n+1$, once the left boundary of one word is equal to the length of $hash\_set['word']$, which means we cannot match sequence B.

By doing this, we finally can work out the maximum venom level of this snake.

The Pseudocode shows below is helpful to understand binary search on $hash\_set['word']$.

Algorithm 1

```
index = −1
for each w in sequence B do
    indexes = hash_set[w]
    left = 0
    right = length of indexes
    while left < right do
        mid = (left + right)//2
        if indexes[mid] > index then
            right = mid else left = mid + 1
        end if
    end while
    if left ==length of indexes then
        return False
    end if
    index = indexes[left]
end for
```

Q2

tutorial 3- 15

Q3

The picture of activities question on lecture

Q4

Q5