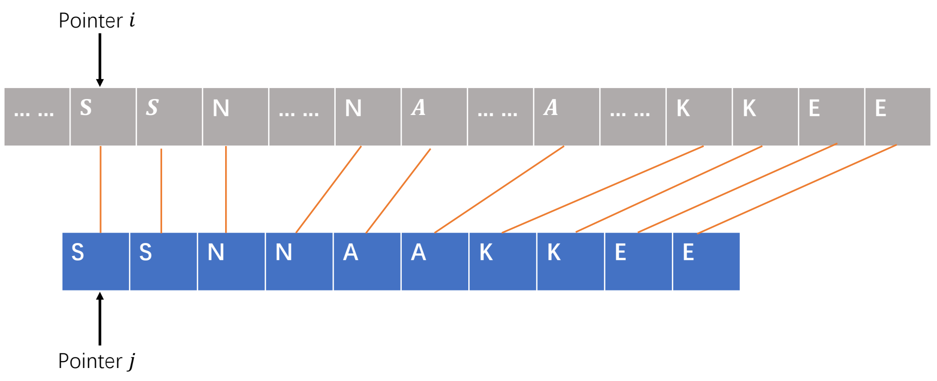
Yuan Gao z5239220 Q1

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.

We can think of this as a substring matching problem:
For instance:

1. We create two pointers i and j, i points to DNA sequence and j points to level of venom sequence.



   If i equal to j, both pointers move right one unit.
   Otherwise, i pointer move right one unit.
   In the end, if pointer j move to the end of venom sequence, that proof this snake has this level of venom. Otherwise, this DNA does not satisfy this level of venom.

2. We can use binary search to select the level of venom.
   a) Because all of words should be covered x times, the maximum level of venom is 1/5 length of DNA sequence.
   b) The minimum level of venom is 0.
   c) If a DNA sequence satisfies x level of venom and it must satisfy $x - 1$ level of venom.

   Therefore, we can set 0 as the left side and maximum level as the right side.

   $$mid = \frac{(left + right + 1)}{2}$$

   If mid level venom is venom, $left = mid$. Otherwise, $right = mid - 1$.

Time complexity:
Every scanning and checking whether x level of venom cost $O(n)$ time complexity.
Create venom level sequence costs $O(n)$ time complexity.

Binary search the aim of level of venom costs $O(logn)$ time complexity.

Therefore, the total time complexity is $O\big(logn(n+n)\big) = O(nlogn)$.

Code:

```
/*match substring of dna O(N)*/
bool isVenom(string dna, string venom){
    int i = 0, j = 0;
    int m = dna.size(), n = venom.size();
    while(i < m && j < n){
        if(dna[i] == venom[j]){
            ++i;
            ++j;
        }else{
            ++i;
        }
    }
    return j == n;
}

/*create level of venomSequence O(5*x) = O(1)*/
string venomLevel(int level){
    string sequence = "SNAKE";
    string venom;
    for(int i = 0; i < 5; ++i){
        for(int j = 0; j < level; ++j){
            venom += sequence[i];
        }
    }
    return venom;
}

/*get Maximum venomLevel O(n/5logn)*/
int getvenomLevel(string dna){
    int maxLevel = dna.size() / 5;
    int l = 0, r = maxLevel;
    while(l < r){
        int mid = (l + r + 1) >> 1;
        //create venom sequence
        string venom = venomLevel(mid);
        //check whether dna sequence satisfy venom sequence
        if(isVenom(dna, venom)){
            l = mid;
        }else{
            r = mid - 1;
        }
    }
    return l;
}
```