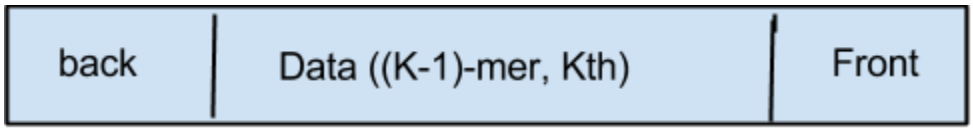
CS 548: Assignment 1

Darshan Washimkar

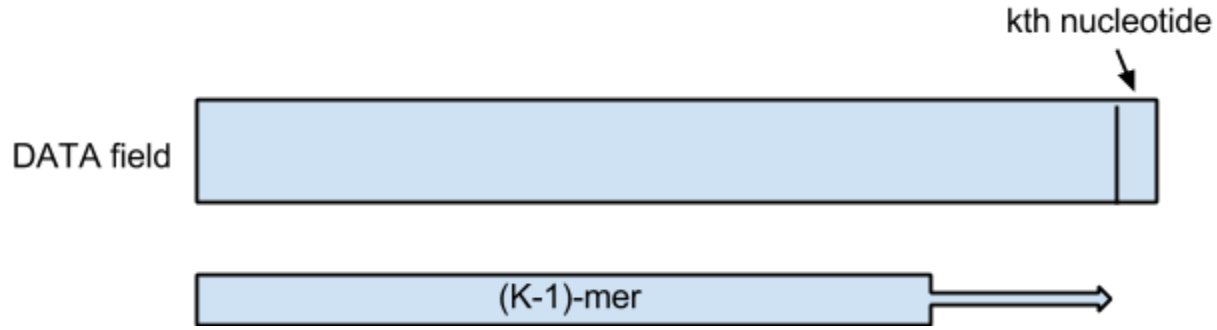
Q1. a. Answer

My program uses **Binary Search Tree,** where each node is as follows



Back and Front are the pointers.
Data field is a string of characters such that it contains

1. (k-1) mer and
2. kth nucleotide of each k-mer P such that (k-1) nucleotides are identical.



```
class kmerStore {
public:
        std::string kmers;
        kmerStore *right;
        kmerStore *left;
}
```

So if some kmer is : **CAGCACA**G
and another kmer is : **CAGCACA**C
Then my data structure will store **CAGCACA**CG, saving space of (k-1) nucleotides.
Please note that kth nucleotide of each kmer is also sorted.

Q1. b. Run time and space required

Lets consider,
Length of each read = *L* (I'm considering read means k-mer)
and Number of **unique** reads = *n*

As I'm using a binary search tree, so the time complexity for my program will be as follows,
For a string, compare function has a time complexity = *O(L)*
and traversing a binary search tree require the complexity of log(n)
also I have (*n*) value to put in to the data structure.

hence the time complexity of my program to create data structure = *O(**L**) O(**nlog(n))***

space complexity = *O(**L**) O(**n**) (more the number of repeated reads less the memory required)*

Q1. c.

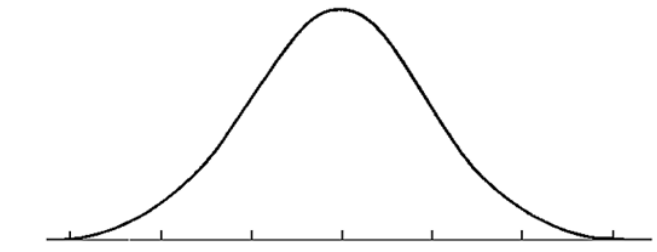
The running time required to return all the adjacent (k−1)-mers for a given(k − 1)-mer
=
The running time required for searching given (k − 1)-mer in Binary Search Tree

= *O**(log n)***

Q3.

For k = 31, Unique k mers = 253957225
For k = 55, Unique k mers = 256180877
For k = 77, Unique k mers = 196132088

As we can see from the different values of k, Unique k mers found for k=55 is highest.
If we draw a graph, of value of **k and number of unique k-mers** then it should look **similar** to the graph shown below.



Value of k Vs number of unique k-mers