Solutions

Anuraag Srivastava(as4378@nau.edu)

1(A). The size of bit array is as shown:

```
[as4378@wind ~ ]$ ./homework hw3_dataset.fa test_genome.fasta A
Initialization started at: 21:5:31
Initialization ended at: 21:5:43
Bit array size in bytes is 536870912
Bit array size in bits is 4.29497e+09
De allocation started at: 21:5:43
Successfully de-allocated the memory used for storing records at: 21:5:43
```

Since there are 4^16 possible combinations for string of length 16 with 4 possible characters, there are in all 4^16 bits to be stored in bit array. Therefore, total number of bytes are 2^29.

1(B). The total number of sequence fragments queried are as shown:

```
[as4378@wind ~ ]$ ./homework hw3_dataset.fa test_genome.fasta B
Initialization started at: 21:5:58
Initialization ended at: 21:6:9
total of 5999245 sequence fragments were queried
De allocation started at: 21:6:9
Successfully de-allocated the memory used for storing records at: 21:6:9
```

We see that that a total of 5999245 sequence fragments were present in the readset. To query the sequence fragment we have to find the radix value for the string representing the fragment. Then we need to find the position for this value in our char array which is found by dividing this value by 8 and checking the numerator. This will give the index in the bit array where the corresponding bit occurs. Then we use the logic in *SetBitArray* function in fastaBA.h file to find the bit position in the corresponding index. After finding the bit we check if the bit is already set by using right shift operator on the value and bitwise AND with 1 (*QueryBitArray* in fastaBA.h file). If the result is 1 it means that bit is already set and we just increment our collision count by 1. Otherwise, we use left-shift on 1 by the corresponding number of bit positions and bitwise OR with the array element to set the bit.

1(C). Total number of unique sequence fragments are as shown:

```
[as4378@wind ~ ]$ ./homework hw3_dataset.fa test_genome.fasta C
Initialization started at: 21:6:24
Initialization ended at: 21:6:35
total of 4414556 unique sequence fragments were found
De allocation started at: 21:6:35
Successfully de-allocated the memory used for storing records at: 21:6:35
```

We see that a total of 4414556 unique sequence fragments are found. The calculation for this is simple, we just need to subtract the number of collisions from total queried elements.

2(A). Total number of collisions when the hash table size is 10,000 are as shown:

```
[as4378@wind ~ ]$ ./homework hw3_dataset.fa test_genome.fasta D
Initialization started at: 20:32:6
Initialization ended at: 20:32:17
total of 5989245 collisions occured
De allocation started at: 20:32:17
Successfully de-allocated the memory used for storing records at: 20:32:17
```

We see that a total of 5989245 collisions occurred and it took about 11 seconds to initialize the hash table. To initialize the hash table we are using chaining. First we declare an array of pointers to Node with following declaration for Node:

struct Node {

unsigned int radix; // the radix value for the genome string

Node* next; // pointer to next node

};

Then we initialize each element to NULL. Then for each fragment in read set we calculate the radix value and find the corresponding index in hash by using division method i.e. (radix value % 10000) to get the corresponding index in the hash table array. Then we create a new Node for this element and add this to the front of the list which takes O (1) complexity.

2(B). The total number of collisions when using hash table of size 100,000 elements is as shown:

```
[as4378@wind ~ ]$ ./homework hw3_dataset.fa test_genome.fasta E
Initialization started at: 20:32:38
Initialization ended at: 20:32:49
total of 5899245 collisions occured
De allocation started at: 20:32:49
Successfully de-allocated the memory used for storing records at: 20:32:49
```

We see that a total of 5899245 collisions occurred and it took about 11 seconds to initialize.

2(C). The total number of collisions when using hash table of size 10,000,000 elements is as shown:

```
[as4378@wind ~ ]$ ./homework hw3_dataset.fa test_genome.fasta F
Initialization started at: 20:33:26
Initialization ended at: 20:33:37
total of 2618870 collisions occured
De allocation started at: 20:33:37
Successfully de-allocated the memory used for storing records at: 20:33:37
```

We see that a total of 2618870 collisions occurred and it took about 11 seconds to initialize.

We see that time taken for initializing is same regardless of the hash table size. This is because we are always adding the element to the front of the list in the hash table which takes O(1) time complexity. Thus if we are querying the same number of elements it would take the same time in all the cases.

2(D). By querying all the possible 16-mers in test_genome.fasta read set the results are as shown:

```
[as4378@wind ~ ]$ ./homework hw3_dataset.fa test_genome.fasta G
Initialization started at: 20:33:49
Initialization ended at: 20:34:1
Searching genome data started at:20:34:1
A total of 5227278 sequence of 16 mers were queried
In all 926494 of 16 mers were found in dataset
Search ended at: 20:34:10
De allocation started at: 20:34:10
Successfully de-allocated the memory used for storing records at: 20:34:10
```

We see that a total of 926494 sequence fragments were found in our hash table. It took about 9 seconds to query all the 16-mers.