Homework #1

Due February 15th, 11:59pm

Each homework submission must include:

- An archive (.zip or .gz) file of the source code containing:
  - The makefile used to compile the code on Monsoon **(5pts)**
  - All .cpp and .h files **(5pts)**
- A full write-up (.pdf of .doc) file containing answers to homework's questions **(5pts)**, including the exact command line needed to execute every subproblem of the homework

The source code must follow the following guidelines:

- No external libraries that implement data structures discussed in class are allowed, unless specifically stated as part of the problem definition. Standard input/output and utilities libraries (e.g. math.h) are ok.
- All external data sources (e.g. input data) must be passed in as a command line argument (no hardcoded paths within the source code **(5pts)**.
- Solutions to sub-problems must be executable separately from each other. For example, via a special flag passed as command line argument **(5pts)**

For this homework, you will need to use the High Throughput Sequence reads dataset located on Monsoon: /common/contrib/classroom/inf503/hw_dataset.fa

- This file contains approximately 36 million 'reads' (genomic sequence fragments of equal length) from multiple datasets (14 in total)
- Each read is exactly 50 nucleotides (characters) long
- The read set is in FASTA format (see insert)
  - The headers are unique and consist of the read ID number (e.g. R1) and a series of 'copy number' values for the number of times this read is present in sample 1, 2, … (separated by underscore "_")
  - The genomic sequences consist of the following alphabet {A, C, G, T, N}

```
>R0_1_0_0_0_0_0_0_0_0_0_0_0_0
GTAACTGAACTGTTTGGTCAGCTCAGCGACTACAGACGACTTGTAGTAAT
>R1_0_0_0_0_0_0_0_0_1_0_0_0_0
AGGGGCAGGCGTACGGCCTTTTCTTCGCGCTCGTCGCGAACGACCGCGCG
>R2_0_0_0_1_0_0_0_1_0_0_0_0_0
AATGGCTTTTTTTCCAAAGATAAACCGAATTTTTTAATATATTTACTGAC
>R3_0_0_0_0_0_0_0_0_0_0_0_1_1
GTGACCCAGAAACCCAACCGATCATGATGCGTCTGCAATCGGATCTGGTT
>R4_1_0_0_0_0_0_0_0_0_0_0_0_0
TTCCGAAAGCTGTACTAAGCCTTTCAGCAGTTGCTTTTGCTTGAGTGGGT
>R5_0_0_0_0_0_0_0_0_0_0_1_0_0
ACGAGAACTGATAGCCGGCCGTCACCGCGACGCGCTGCTCCGCCTGCGCG
>R6_0_0_0_0_1_0_0_0_0_0_0_0_0
TGTCGAAGGATGTCGGTAAATCGATATTCTGTGTCGAAACGTCGATATAA
>R7_0_4_0_0_0_0_0_0_0_0_0_0_0
```

Problem #1 (of 1): Arrays and Classes

Create a class called ***FASTA_readset***. The purpose of the class will be to contain **a single** FASTA read dataset (so you'll need 14 instances of this object) and all of the functions needed to operate on this set. Use an array data-structure to store the genomic sequence of the given read dataset. Use character arrays (char[ ]) to store the sequence, rather than 'string' object (you should have an array-of-arrays object to store a single dataset). At minimum, the class must contain (**15pts**):

- A default constructor (zeroes everything out)
- At least one custom constructor (parses the combined file and fills in the actual data)
- A function to alphabetically sort the sequence fragments within the FASTA_readset
- A function to implement a binary search within the fragments of the FASTA_readset
- A single function to compute the statistics for the Readset (see below)
- A destructor
- Comments describing major code blocks and control structures

A. **(20pts)** Read in the combined dataset and initialize all 14 instances of the FASTA_readset object. Hint: You may want to retain the copy count of each fragment as a separate array.

   - How many unique sequence fragments are in each of the 14 datasets?
   - How many total sequence fragments are in each dataset (i.e. when you consider copy numbers)?

B. **(20pts)** Without alphabetically sorting any of the data in the FASTA_readset object compare the contents of datasets 1 and 2 (i.e. use the fragments in dataset 1 as queries to search in dataset 2). Make sure you continue to consider copy count in your answer.

   - What is the 'big O' notation of your search (linear / quadratic / cubic / etc)?
   - How long does it take (in seconds) to search for all fragments of dataset 1 within dataset 2? Please note that depending on the efficiency of your algorithm, this step may take a long time. First estimate the total time using 1,000, 10,000, and 100,000 queries – if total time estimate is greater than 24 CPU hours, provide estimate rather than exact number.
   - How many sequence fragments in dataset 1 are also in dataset 2? (estimate if needed)

C. **(20pts)** Alphabetically sort the sequence fragments in each of the FASTA_readset objects and implement a binary search function to compare the contents of datasets 1 and 2 (i.e. use the fragments in dataset 1 as queries to search in dataset 2).

   - What is the 'big O' notation of your search (linear / quadratic / cubic / etc)?
   - How long (in seconds) does it take to search for 1000 queries? How about 10,000 or 100,000? Does the time increase make sense? Explain the differences (if any) when compared to search times obtained as part of 1B.

   - How many sequence fragments in dataset 1 are also in dataset 2?