

EXERCISES

3.1.1 Write a client that creates a symbol table mapping letter grades to numerical scores, as in the table below, then reads from standard input a list of letter grades and computes and prints the GPA (the average of the numbers corresponding to the grades).

A+	A	A–	B+	B	B–	C+	C	C–	D	F
4.33	4.00	3.67	3.33	3.00	2.67	2.33	2.00	1.67	1.00	0.00

3.1.2 Develop a symbol-table implementation `ArrayST` that uses an (unordered) array as the underlying data structure to implement our basic symbol-table API.

3.1.3 Develop a symbol-table implementation `OrderedSequentialSearchST` that uses an ordered linked list as the underlying data structure to implement our ordered symbol-table API.

3.1.4 Develop `Time` and `Event` ADTs that allow processing of data as in the example illustrated on page 367.

3.1.5 Implement `size()`, `delete()`, and `keys()` for `SequentialSearchST`.

3.1.6 Give the number of calls to `put()` and `get()` issued by `FrequencyCounter`, as a function of the number W of words and the number D of distinct words in the input.

3.1.7 What is the average number of distinct keys that `FrequencyCounter` will find among N random nonnegative integers less than 1,000, for $N=10$, 10^2 , 10^3 , 10^4 , 10^5 , and 10^6 ?

3.1.8 What is the most frequently used word of ten letters or more in *Tale of Two Cities*?

3.1.9 Add code to `FrequencyCounter` to keep track of the *last* call to `put()`. Print the last word inserted and the number of words that were processed in the input stream prior to this insertion. Run your program for `ta1e.txt` with length cutoffs 1, 8, and 10.

3.1.10 Give a trace of the process of inserting the keys `E A S Y Q U E S T I O N` into an initially empty table using `SequentialSearchST`. How many compares are involved?

3.1.11 Give a trace of the process of inserting the keys `E A S Y Q U E S T I O N` into an initially empty table using `BinarySearchST`. How many compares are involved?

3.1.12 Modify `BinarySearchST` to maintain one array of `Item` objects that contain keys and values, rather than two parallel arrays. Add a constructor that takes an array of

EXERCISES *(continued)*

Item values as argument and uses mergesort to sort the array.

3.1.13 Which of the symbol-table implementations in this section would you use for an application that does 10^3 put() operations and 10^6 get() operations, randomly intermixed? Justify your answer.

3.1.14 Which of the symbol-table implementations in this section would you use for an application that does 10^6 put() operations and 10^3 get() operations, randomly intermixed? Justify your answer.

3.1.15 Assume that searches are 1,000 times more frequent than insertions for a BinarySearchST client. Estimate the percentage of the total time that is devoted to insertions, when the number of searches is 10^3 , 10^6 , and 10^9 .

3.1.16 Implement the delete() method for BinarySearchST.

3.1.17 Implement the floor() method for BinarySearchST.

3.1.18 Prove that the rank() method in BinarySearchST is correct.

3.1.19 Modify FrequencyCounter to print all of the values having the highest frequency of occurrence, not just one of them. *Hint*: Use a Queue.

3.1.20 Complete the proof of PROPOSITION B (show that it holds for all values of N). *Hint*: Start by showing that $C(N)$ is monotonic: $C(N) \leq C(N+1)$ for all $N > 0$.