CSCI 340 Fall 2023 Section 1
Test 1 review sheet

**git/github**

What is git for?
What are different branches for?
What is a commit?
Common git commands:

git clone
git add
git commit
git push

***Standard Template Library***

Containers

vector: sequence container

What are the common methods? begin()/end()/push_back()
Is the method size() same as capacity()?

list: sequence container
deque:
map: associative container; sorted; unique.

The elements in a map is of the type *pair<T1, T2>.*

set: associative container; sorted; unique
stack: container adaptor; FILO
queue: container adaptor: FIFO

There are other containers in STL (e.g. multimap which allow duplicates). You are required to memorize the above containers' important usages, such as **how to create, how to iterate through, and how to access**.

Give examples of methods that are defined in list but not in vector (push_front)? Or vice versa (at)?

Examples:

1. How to declare a vector of five 0s?
2. Does list have at() method?
3. What is the result of codes below?
   *map<const char*, int, strCmp> ages;*
   *ages["Homer"] = 38;*
   *ages["Homer"] ++;*
4. What are the variable names for the two objects stored in a pair<T1, T2>? Answer: first, second.

Iterators

What is an iterator?
When do you need const_iterator? (The values it points to can't be changed.)
What are different categories of iterators?

InputIterator; OutputIterator; ForwardIterator; BidirectionalIterator, RandomAcessIterator

What are iterator adaptors?

Reverse iterators; insert iterators; stream iterators

How to use an iterator to iterate through a vector?
How to use an iterator to iterate through a map?

Algorithms

There are 4 types of STL algorithms:

Nonmodifying algorithms (find, count etc.)

Modifying algorithms (generate, unique, replace, copy etc.)
Numeric algorithms (inner_product etc.)
Sorting algorithm (sort, heap operations etc.)

In particular:

- find(begin, end, value): //return an iterator of position of value in the unsorted sequence; if not present, return end;
- sort(begin, end); //sort the sequence into ascending order
- count(begin, end, value); //return how many times value occurs in the sequence.
- for_each(being, end, function);
- High order function: function as an argument for STL algorithms.

Extra example of modifying algorithm: generate(); generate_n():

- Generate elements and fill a sequence between a set of iterators (generate) or from a starting point through the next n elements (generate_n). The values that are used to fill a sequence are created by a function.

```
void generate(start_iterator, end_iterator, function_that_generates_values);
```

### *Complexity analysis*

1. Efficiency of algorithms are defined in terms of how running times scale as a function of the size of the input (also called asymptotic complexity).
2. Big-O: Defines an estimated upper bound for the running time.

3. Understand the relationship among different functions:
   e.g.: $4n+5 = O(n)$ -> eliminate the lower order tem or constant coefficients
   $n^2 + n = O(n^2)$
   It is a one-way equality!
4. Analysis of codes.
   Complexity analysis of codes with loops
   Understand the codes of the maximum subsequence sum --- why the implementations are of certain complexity? O(n^3); O(n^2); O(nlogn); O(n)

5. Common complexity: linear search? Binary search? Random access using []? push_back() for vector?

Example: Analyze the complexity of the following code segment using big-Oh notation

```
for i = 1 to n do
  for j = i to n do
    sum = sum + 1
```

Answer: $T(n) = n + (n-1) + .. + 1 = n(n+1)/2 = O(n^2)$) ( To be specific, it is actually a tight bound, which can be noted as $\theta (n^2)$. But you can just say $O(n^2)$)

### *Recursion*

What is an anchor case (base case)?
What if a recursive function does not have an anchor case?
What is an activation record?

Can you summarize some key points of writing recursive function?
What are the situations where a recursion function is suitable?
How to write a simple recursive function given a recursive definition?

### **Miscellaneous**

Pass by value /pass by reference
How to write template functions

***Assignments 1 2(2a), 2b, 3***

Assignment 1

Git commands

Assignment 2a

Use an iterator to access a container, and do some statistical summary, histogram, and tokenization

Assignment 2b

Write a simple linked list class, and its iterator

Assignment 3

Access a vector to provide 2D access, and do matrix multiplication

***Dailies***

Some basic logic may be tested.