

Homework 1

Data Structures and Object-oriented Programming

Instructor: Feng-Jian Wang

Due: April 8, 2021

- The STL Algorithm *count* which has the syntax
$$\text{count}(\text{start}, \text{end}, \text{value})$$
returns the number of occurrences of value in the range [start, end]. Write a program that uses this algorithm to determine the number of occurrences of a[0] in the integer array a[0:n-1]. Test your code.
 - Write a function for Matrix transpose.
- Show that the following equalities are correct.
 - $$\sum_{i=0}^n i^3 = \Theta(n^4)$$
 - $$n^{2^n} + 6 \cdot 2^n = \Theta(2^{2^n})$$
 - $$33n^3 + 4n^2 = \Omega(n^2)$$
- Show that the following equalities are incorrect.
 - $$n^2 \log n = \Theta(n^2)$$
 - $$n^3 2^n + 6n^2 3^n = O(n^3 2^n)$$
- Implement a class *Complex*, which represents the Complex Number data type. Implement a constructor (including a default constructor which creates the complex number 0 + 0i).
- Implement a class *Quadratic* that represent 2-degree polynomials i.e., polynomials of type $ax^2 + bx + c$. Your class will require three data members corresponding to a, b and c. Implement the following operation:
Overload **operator+** to add polynomials of degree 2.
- Implement *Queue* as a public derived class of *Bag* using templates.
- Write a C++ function to transform from prefix to postfix. Carefully state any assumptions you make regarding the input. How much time and space does your function take?
- What is the prefix form of the expression $A * B * C$?
- Write a C++ function, **operator*(const Matrix &b) const**, which returns the matrix ***this * b**. If a is an $n \times m$ matrix with r_a nonzero terms and if ***this** is an $n \times m$ matrix with r_a nonzero terms and b is an $m \times p$ matrix with r_b nonzero terms, then this multiplication can be done in $O(pr_a + nr_b)$ time. Can you think of a way to do the multiplication in $O(\min\{pr_a, nr_b\})$ time?

10. Write and test a copy constructor for sparse matrices. What is the computing time of our copy constructor?