Zachary Qian

CS 32 Spring 2019

Homework 4

2. The one argument form of the insert function for the coordinate template class has a compile error because the template’s parameter/argument deduction was ambiguous or not specified. Because the one argument insert function using the Coord template class requires a comparison, we are required to have a “>” operator overload. The one argument insert function using an integer variable type is valid because C++ can already deduce and compare int types without our own operator overload due to its primitive nature. Thus, because C++ cannot compare the two Coord objects and thus cannot insert it at the “proper” place without a “>” overload, there is a compiler error.

4b. Given the constraints from 4a, we would need two parameters because we need to both maintain and remember the prior updates of the labels of the domain and subdomains, while also accessing the future subdomains up to the leaf node. A one parameter overload does not suffice because of the need to track prior labels, as well as the need to access the future labels of the child nodes. (subdomains) of the given domain/subdomain.

5a. The complexity of this algorithm of mutual friends is O(N3). Each loop will run through N times each, both worst case and on average. Because the innermost for loop is nested within another nested loop, it is N raised to the third power. Each loop operates through N times, thus making the overall complexity of this algorithm O(N3).

5b. The complexity of the newer mutual friends algorithm remains O(N3). This is because the worst-case scenario, such that i is equal to N, makes the algorithm remain O(N3). The average case scenario of the first two for loops would be reduced to O(0.5N2). However, the first loop would loop N times still, leading to an overall average complexity of O(0.5N3). This, in the long term, would disregard the coefficient, thereby making the complexity remain at O(N3) after combining the other loops.

6a. The complexity of the interleave algorithm is O(N2). This is because the get accessor method is of complexity O(N), as it, on average, runs through the linked list N iterations. Because the get method is used within a for loop that runs N times, the complexity of the first loop is O(N2). The next for loop is also of complexity O(N2). Because the loop also employs the get method, which is O(N), the combination of the loop and the nested get method results in a complexity of O(N2). Thus, the complete complexity is O(N2+ N2) = O(2 N2). However, we can disregard the lower order monomial, making the interleave algorithm have the complexity of is O(N2). If we want to explore the technicalities, the get method is used for both sequences, meaning the complexity within the first loop is O(2N).

6b. The complexity of the newer interleave algorithm is of complexity O(N). Within the first loop, there are only insertBefore() methods. Because insertBefore does not need a comparison and only identifies a valid location to insert the value, it is of complexity O(1). This follows for both insertBefore() statements. Thus, the complexity of the first loop is O(N), as we disregard coefficients. For the second loop, there is only one insertBefore, and the loop, at worst, will loop. N times. Thus, the complexity of the second loop is also O(N). Finally, the swap function goes through the linked list N times as well. Thus, because we disregard the leading coefficient, the complexity of the entire algorithm is still O(N).