CS300 Project 1

Big O Algorithm Analysis

Jared Hodgkins

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Worst Cost** | **SUM Worst Cost** | **Worst Case** |
| Course() | 3 | 3 | O(1) |
|  |  |  |  |
| Parser::Parser() | 4N + 11 + Parser::parseContent() | 9N + 4CN +13 | O(N) |
| Parser::parseContent() | 5N + 4CN + 2, N(5 + 4C) + 2 | 5N + 4CN + 2 | O(N) |
| Parser::validatePrerequisite() | 2N + 2 | 2N + 2 | O(N) |
| printCourse() | 7 | 7 | O(1) |
|  |  |  |  |
| partition() "vector<Course>" | 5\*(N/2) + 6, 5N/10 + 6 | N/2 + 6 | O(N) |
| partition() "vector<string>" | 5\*(N/2) + 6, 5N/10 + 6 | N/2 + 6 | O(N) |
| quickSort() "vector<Course>" | N\*partition() + 6 | N^2/2 + 12 | O(N^2) |
| quickSort() "vector<string>" | N\*partition() + 6 | N^2/2 + 12 | O(N^2) |
|  |  |  |  |
| loadCoursesVect() | 5N + 3PN + 4, N(5 + 3P) + 4 | 5N + 3PN + 4 | O(N) |
| findCourseVect() | 2N + 2 | 2N + 2 | O(N) |
| printCoursesVect() | 2N \* printCourse() | 14N | O(N) |
|  |  |  |  |
| HashTable::InitNodes() | 2 \* tableSize | ~ 200 | O(1) |
| HashTable::HashTable() | HashTable::InitNodes() + 3 | 2 \* tableSize + 3 | O(1) |
| HashTable::hash() | 2 \* num\_chars + 2 | ~16 | O(1) |
| HashTable::Search() | 4N + 3 | 4N + 3 | O(N) |
| HashTable::Insert() | 2N + 8 + HashTable::hash() + HashTable::Search() | ~ 6N + 27 | O(N) |
| loadCoursesHashTable() | HashTable() + Parser() + 6N + 2 | 155N + 4CN + 2t + 7 | O(N) |
| HashTable::Keys() | 7N + 1 | 7N + 1 | O(N) |
| printCoursesHT() | Keys() + quickSort() + N\*HashTable::Search() + N\*printCourse() | 4.5N^2 + 17N + 13 | O(N^2) |