**Runtime Analysis**

|  |  |  |  |
| --- | --- | --- | --- |
| **Operation** | **Vector** | **Hash Table** | **Binary Search Tree** |
| **Open File** | Cost: 1 | Cost: 1 | Cost: 1 |
|  | Executions: 1 | Executions: 1 | Executions: 1 |
|  | Total Cost: O(1) | Total Cost: O(1) | Total Cost: O(1) |
| **Read Line** | Cost: 1 | Cost: 1 | Cost: 1 |
|  | Executions: n | Executions: n | Executions: n |
|  | Total Cost: O(n) | Total Cost: O(n) | Total Cost: O(n) |
| **Split Line into Tokens** | Cost: O(m) | Cost: O(m) | Cost: O(m) |
|  | Executions: n | Executions: n | Executions: n |
|  | Total Cost: O(n\*m) | Total Cost: O(n\*m) | Total Cost: O(n\*m) |
| **Check Formatting of Tokens (size)** | Cost: 1 | Cost: 1 | Cost: 1 |
|  | Executions: n | Executions: n | Executions: n |
|  | Total Cost: O(n) | Total Cost: O(n) | Total Cost: O(n) |
| **Create Course Object** | Cost: 1 | Cost: 1 | Cost: 1 |
|  | Executions: n | Executions: n | Executions: n |
|  | Total Cost: O(n) | Total Cost: O(n) | Total Cost: O(n) |
| **Add Prerequisites (Loop through tokens)** | Cost: 1 | - | - |
|  | Executions: n (max k times) | - | - |
|  | Total Cost: O(n\*k) | - | - |
| **Compute Hash Index** | - | Cost: 1 | - |
|  | - | Executions: n | - |
|  | - | Total Cost: O(n) | - |
| **Insert Course into Hash Table** | - | Cost: 1 (avg case) | - |
|  | - | Executions: n | - |
|  | - | Total Cost: O(n) | - |
| **Insert Course into BST** | - | - | Cost: O(h) (h is height) |
|  | - | - | Executions: n |
|  | - | - | Total Cost: O(n^2) (worst case) or O(n log n) |

|  |  |  |
| --- | --- | --- |
| Data Structure | Total Cost | Worst-case Big O Value |
| Vector | O(n\*m+n\*k) | O(n\*m) |
| Hash Table | O(n\*m | O(n\*m) |
| Binary Search tree | O(n\*m+n^2) | O(n^2) |

**Explanation of the Tables:**

Table 1: Provides a detailed breakdown of each operation across the three data structures, showing the cost per line, number of executions, and the resulting total cost for each operation separately.

Table 2: Summarizes the cumulative costs of loading courses and the worst-case Big O values for each data structure, allowing a clear comparison of their overall performance.