

The following runtime tables evaluate the worst-case cost for reading the course file and constructing each data structure.

#### Vector Runtime Analysis

Operation analyzed: reading the CSV file and loading all Course objects into a vector<Course>.

Code	Line Cost	# Times Executes	Total Cost
create vector	1	1	1
open file	1	1	1
for each line in file	1	n	n
parse line into tokens	1	n	n
create Course object	1	n	n
add each prerequisite to list	1	n	n
append Course to vector	1	n	n
close file	1	1	1
Total Cost			$5n + 3$
Runtime			$O(n)$

#### Hash Table Runtime Analysis

Operation analyzed: reading the CSV file and loading all Course objects into a hash table keyed by courseNumber. The average case would be closer to  $O(n)$  but we are being asked WORST CASE.

Code	Line Cost	# Times Executes	Total Cost
create hash table	1	1	1
open file	1	1	1
for each line in file	1	n	n
parse line into tokens	1	n	n
create Course object	1	n	n
add each prerequisite to list	1	n	n
compute hash key for courseNumber	1	n	n
insert Course into hash table (worst case)	n	n	$n^2$
close file	1	1	1
Total Cost			$n^2 + 5n + 3$
Runtime			$O(n^2)$

#### Binary Search Tree (BST) Runtime Analysis

Operation analyzed: reading the CSV file and inserting all Course objects into a binary search tree ordered by courseNumber.

Code	Line Cost	# Times Executes	Total Cost
create empty tree	1	1	1
open file	1	1	1
for each line in file	1	n	n
parse line into tokens	1	n	n
create Course object	1	n	n
add each prerequisite to list	1	n	n
insert Course into tree (worst case)	n	n	$n^2$
close file	1	1	1
Total Cost			$n^2 + 4n + 3$
Runtime			$O(n^2)$