**Runtime Analysis for a Vector**

Open file: O(1)

FOR EACH line (loop executes n times):

SPLIT line: O(m) (m is the average number of components per line)

Course object creation: O(1)

Add to vector: O(1)

Close file: O(1)

Overall runtime: O(n \* m)

**Memory Usage for a Vector**

Vector: stores n courses

Each course requires memory proportional to the number of characters in the course number, name, and prerequisites

Memory usage: O(n \* m)

**Runtime Analysis for a Hash Table**

Open file: O(1)

FOR EACH line (loop executes n times)

SPLIT line: O(m)

Course object creation: O(1)

Add to hash table: O(1)

Close file: O(1)

Overall runtime: O(n \* m)

**Memory Usage for a Hash Table**

Hashtable: stores n courses

Each course requires memory proportional to the number of characters in the course number, name, and prerequisites

Memory usage: O(n \* m)

**Runtime Analysis for a Binary Search Tree**

Open file: O(1)

FOR EACH line (loop executes n times)

SPLIT line: O(m)

Course object creation: O(1)

Insert into BST: O(log n) (assume balanced BST)

Close file: O(1)

Overall runtime: O(n \* (m + log n))

**Memory Usage for a Binary Search Tree**

BST: stores n courses

Each course requires memory proportional to the number of characters in the course number, name, and prerequisites.

Memory usage: O(n \* m)