Grant Mercer

Assignment 13

Part A

|  |
| --- |
| **HashedGraph** |
| -hashedList: HashList |
| -vertexCount: int |
| -title: string |
| -edgeCount: int |
| +HashedGraph() |
| +~HashedGraph |
| +addVertex(const string&): string |
| +addEdge(const string&): string |
| +getBaconNumber(const string&) const: string |

|  |
| --- |
| HashList |
| -hashTable: \*int |
| -loadFactor=0.65: static const double |
| -initialTableSize=503: static const int |
| +HashList() |
| +~HashList()` |
| +insert(cons int): bool |
| +remove(cons tint): bool |
| -rehash(): void |
| -hash(const int): int |

**Data structures proposed for the graph data and explain why that data structure was chosen**

I chose to use two important data structures for my assignment. For storage I chose to implement a hash table in order to store the names or actors and movies for constant lookup and storage. The hash table will allow me to correctly store nodes into a table and also lookup when performing operations. The other data structure I’m using is an adjacency list that wraps around the hash table. The adjacency list will hold the connections in the graph allows me to find the kevin bacon number of each actor.

**Finding the Kevin bacon number**

For finding the Kevin bacon number I plan to create an adjacency list based off all of the connections in the graph. From the adjacency list I will use a breadth first search approach in order to find connections based off of a center node Kevin bacon. This will give me the correct number towards the Kevin bacon number.

**Big-O for single graph search and Big-O for Kevin Bacon numbers generation**

Single graph search: O(|V|)

Kevin Bacon Number Generation: O(|E| + |V|)