**NAME: AMARKAI WANSA AMARTEIFIO**

**Final Project**

**EVALUATION**

* **Vector**:
  + Loading courses: O(n)
  + Printing sorted list: O(n log n)
  + Memory: O(n)
* **Hash Table**:
  + Loading courses: O(n)
  + Printing sorted list: O(n log n)
  + Memory: O(n)
* **Tree (assuming a balanced BST)**:
  + Loading courses: O(n log n)
  + Printing sorted list: O(n)
  + Memory: O(n)

This runtime analysis is well-structured and concise. Each data structure has its own advantages depending on whether sorting is necessary and how retrieval is handled. The vector requires sorting before printing, making its overall complexity O(n log n) for that operation. Meanwhile, the hash table provides efficient data insertion but does not maintain order, requiring additional sorting when printing. Lastly, the balanced BST naturally preserves order and allows efficient retrieval in O(n) without explicit sorting.