A screenshot of a document

Description automatically generated

**Advantages and Disadvantages**

The primary advantage of using a vector data structure is that it is relatively simple to create, update, and manipulate. For this situation, where there aren’t many data inputs, that means that a vector data structure will perform well, without having a large burden on the system.

The primary disadvantage of using a vector data structure is that it will not scale well as the number of courses gets really large. Over time, as courses are added to the vector, performance will begin to decline. To searching for individual courses in particular will become an issue, because it will need to search through the entire vector to find the right one.

The primary advantage of a hash table is the manner in which data in stored lends itself to better perform when the data set is larger. While the dataset is small, hash tables do not necessarily perform better than the other data structures. However, as new courses are added to the dataset, the hash tables performance compared to the others will increase.

The primary disadvantage of a hash table in this particular situation is that it can be a burden to sort the classes into alphanumeric order. The reason for this is that the system would need to first pull all of the nodes out before they can be sorted, then it would be able to replace them in alphanumeric order.

The primary advantage of the binary search tree is that the way the data is stored can allow for efficient sorting and searching in order. This would begin to outperform the vector data structure significantly and allow for easier alphanumeric printing.

The primary disadvantage of the binary search tree is that there is a chance that it becomes unbalanced. This could have major implications both in performance and system burden. Therefore, additional resources would need to be dedicated to ensuring this is not happening.

**Recommendation**

Looking at the runtime analysis, we can see that all the data structures would have the same runtime O(n), because they are all straightforward in their execution and dependent on the number of courses that are fed into the system. However, the vector data structure has a significantly lower total cost as compared to the hash table and binary search tree. That is because both of the latter data structures have a more sophisticated method of storing data. The has table would likely perform the best when searching for information on specific courses due to the nature of how the data is stored, however the burden on the system would still be greater than the vector data structure. Without more information on the importance of performance versus system burden, I’ll assume that the small difference in performance is outweighed by the significant difference in burden. Therefore, I would recommend using the vector data structure, and potentially updating to a hash table when the number of courses makes a large difference in performance.