Module 8: Final Portfolio

Patrick Davis

Colorado State University - Global Campus

CSC506

11-29-2024

**Module 8: Final Portfolio**

As anyone who has worked in the restaurant industry can tell you, the people who choose to work in it often have varying work availabilities. For a floor manager, attempting to create a schedule for who can work what shift can sometime be tedious and more time consuming than it should be. This problem may even extend out to other businesses and industries. To fix this issue, the portfolio project option of job scheduling was selected.

**Project Plan**

Designing the project plan that would be followed was the first step. It was decided that the project plan would be designed in a waterfall approach due to the time frame being short and due to the scope being clearly defined in the beginning. Following this, the structured steps of the project would need to align with the milestones set in the class. This meant that project would be split into six stages of 1. Planning and Research, 2. Algorithm Implementation and Testing, 3. Refining and Collecting Data, 4. Analysis and Optimization, 5. Final Refinement, and 6. Testing and Submission.

**Design, Analysis, and Refinement**

The program was initially designed to test three different sorting methods using a greedy, a round robin, and an EDF algorithm. The early designs of the program were geared towards testing with the methods of creating employee objects that would be sorted to be randomized (see appendix A, image 1). The creation of shifts for each day were randomized as well (see appendix A, image 2). This randomization of employees and shifts was done for testing purposes to create a more diverse set of testing data. Another feature worth mentioning is that if any of the sort were unable to place an employee into a shift, that shift would be marked as “Urgent”.

Testing was performed and the result of different test showed that all three algorithms were able to complete the task with the same results (see appendix A, image 3, 4, and 5). This meant that any algorithm would work within the scope of the project. Based on this, the selected algorithm was the greedy algorithm as of the three it was the simplest to modify and work with. Testing had also revealed that while the sorting, each of the algorithms were sorting through the entire list of employees, increasing the run time unnecessarily. Furthermore, randomizing the order the employees stored in the data structure the sorting accessed was leading to some unfair sorting.

During the refinement phase of the project, the issues of the algorithm searching the entire employee database and the unfair sorting were fixed. The first issue of the database search was handled by creating a function that would build a map of the available employees for a certain shift. This way the sort only checks what employees can work that shift rather than the entire database. Fixing the randomness of the sorting did create a new issue. Prior to the change, the sort had a run time of O(n) with n being the number of people in the list. The change that was implemented was that the employee map would be sorted based on the number of shifts each employee had in ascending order and each time the sorting algorithm performed a sort, the map would be re-sorted.

While this change would make the sorting more deterministic in its approach, it changed the run time from O(n) to O(w \* d \* s \*log n) with “w” being the number of weeks, “d” being the number of days, and “s” being the number of shifts. Ultimately, due to this project being created with small businesses in mind, this time change was acceptable. From here, all that was left to do was configure to program to take user input for creating employes.

Conclusion

Developing the job sorting algorithm has been a challenging and fruitful experience. The project allowed for growth in project planning, and algorithm analysis. It has provided not only many lessons, but a project that can be taken forward out of the class and into the real world.

Appendix A

Image 1. Showcase how employees are created with random characteristics.

A screen shot of a computer code

Description automatically generated

Image 2. Showcase how shifts are designed with random characteristics.

A computer code on a black background

Description automatically generated

Image 3.

A screenshot of a computer program

Description automatically generated

Image 4.

A screenshot of a computer program

Description automatically generated

Image 5.

A screenshot of a computer program

Description automatically generated