

Project 1

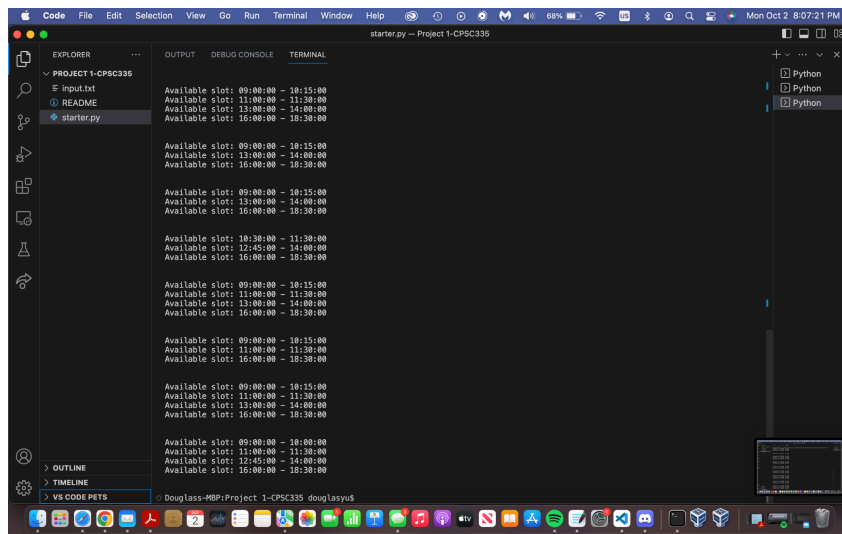
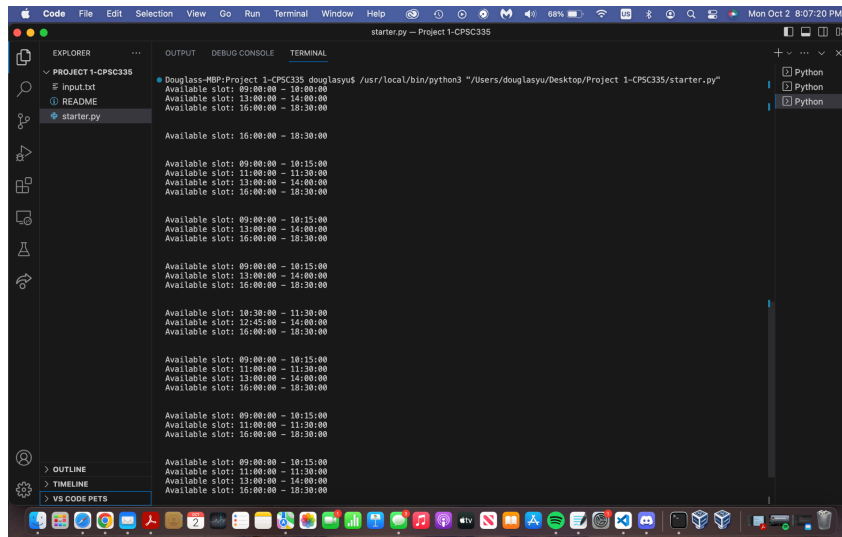
Douglas Yu (douglasmyu@csu.fullerton.edu), Shadi Hirbawi
(hirbawishadi@csu.fullerton.edu), Taylor Livingston (livingtay77@csu.fullerton.edu)

CPSC 335 - Algorithm Engineering

Professor Ornella Irene Dsouza

2 October, 2023

Screenshot of Output



Github link:

<https://github.com/Douglasmyu/groupscheduling>

Time complexity Proof:

When the main function runs, the compiler reading the input takes a time complexity of $O(n)$ time. With the function 'parse_time_range', it is called to split the string by ':' to get

the start and end times. In order to fulfill this task, it is called every time in the for loop during the function 'scheduleing' but only performs one task. Due to this, the time complexity of the function 'parse_time_range' is $O(1)$. When creating each list, each line is run through linear time therefore resulting in a time complexity of $O(1)$. As the program progresses with sorting the meeting schedules and times, there is a nested for loop that not only sorts the schedules but combines them as well into one list. With this, the time complexity results as $O(n(\log n))$ time. Overall, the program runs with a time complexity of $O(n(\log n))$.

Mathematical Analysis:

Parse_time_range function: $O(r + 2) \rightarrow O(r)$

Split time_range = $O(r)$

Strptime = $O(1)$

Strptime = $O(1)$

Scheduling function: $O(n(\log n)) \rightarrow$ drop the constants

For meetHrs1_list = $O(n)$

For meetHrs1_list = $O(n)$

For wrkHrs1_list = $O(n)$

For wrkHrs1_list = $O(n)$

If else statements \rightarrow Range_work.append = $O(1)$

Creating meeting duration = $O(1)$

Combining the schedules: $O(n + n) = O(2n)$

meetHrs1_list = $O(n)$

meetHrs2_list = $O(n)$

Sorting \rightarrow using nested loops with for and if = $O(2n * (\log(2n)))$

Initializing start_time = $O(1)$

For combined_schedule = $O(2n)$

If statement = $O(1)$

If statement = $O(1)$

Get start_time and end_time = $O(1)$

For available slots: $O(2n)$

Printing the string = $O(1)$

Main function: $O(n(\log n))$

Opening file and reading data = $O(d)$

For loop = $O(y)$

If statements = $O(i)$

Scheduling function = $O(n(\log n))$

Printing the newline = $O(1)$