BLG 336E - Analysis of Algorithms 2
Homework #3

Due: 14.05.24, 23:59

Introduction

You are assigned to find solutions to some problems that occurred during the move of Faculty of Computer and Informatics Engineering to its new building. Schedules should be created for the renovations. You also need to make some purchases with the budget provided to the faculty.

For any issues regarding the assignment, please contact Burcu Kartal (kartalbu20@itu.edu.tr).

Implementation Notes

The implementation details below are included in the grading:

1. Follow a consistent coding style (indentation, variable names, etc.).
2. Divide your code into functions and write comments that are descriptive.
3. Be careful about any possible memory leaks and invalid memory use. Check your code with Valgrind.
4. Test your code with given test cases using Calico.
5. Make sure your code is running in the provided environment.

Submission Notes

1. Write your name and ID on the top of the file(s) that you will upload as in the following format:

```
/* @Author
* Student Name
* Student ID */
```

2. Submissions are made through only the Ninova system and have a strict deadline. Assignments submitted after the deadline will not be accepted. You are suggested to submit your progress regularly.

3. This is not a group assignment and getting involved in any kind of cheating is subject to disciplinary actions.

1. Scheduling and Purchasing (60 Pts.)

During the moving of the faculty, it was determined that there were many renovations that needed to be done in the new building, and some of these works to be carried out in the rooms must be planned daily. You are given the time intervals when the rooms on each floor are available. Also, rooms have values that indicate their priorities depending on some factors such as being a professor's office, a laboratory, or a classroom. In addition, workers have to be in the same floor for the whole day all together. And they cannot split and use given time intervals, they can only be in one room during in an interval. Schedules should be created for each floor to maximize the priority gain.

In addition, the rectorate provided 200.000 TL budget to the faculty to purchase new items/devices that were determined to be broken or old while moving the building. Each item that can be purchased has a price and a value depending on its necessity level. You are expected to make the most valuable purchases with the budget.

You should use of the Weighted Interval Scheduling and Knapsack problems to solve the given tasks and consider dynamic programming approach. So, in short, you should find the best possible schedule for each floor and the most valuable purchases you can make for the new building.

You will use 3 input files for each three test cases. The file names and their contents are given below:

- room_time_intervals.txt: Floor Name, Room Number, Start Time, End Time
- priority.txt: Floor Name, Room Number, Priority Value
- items.txt: Item Name, Price, Need Value

The format of a sample output is given below:

```
Best Schedule for Each Floor

Floor_1 --> Priority Gain: 320
Floor_1   Room_1 07:00   09:00
Floor_1   Room_2 09:00   11:30
Floor_1   Room_2 12:00   14:30
…

Floor_2 --> Priority Gain: 230
Floor_2   Room_3 09:30   12:45
Floor_2   Room_2 13:00   14:30
…

Floor_3 --> Priority Gain: 255
Floor_3   Room_4 10:00   11:00
…

Best Use of Budget
Total Value --> 18.3

Computer_1
Chair_3
Desk_2
…
```

2. Report (40 Pts.)

Prepare a report and discuss the following items:

- Explain your code and your solution by:
  - Writing pseudo-code for your functions following the pseudo-code conventions given in the class slides.
  - Show the time complexity of your functions on the pseudo-code.

- What are the factors that affect the performance of the algorithm you developed using the dynamic programming approach?
- What are the differences between Dynamic Programming and Greedy Approach? What are the advantages of dynamic programming?

3. Test

3.1 Valgrind

Valgrind is a tool for memory debugging and memory leak detection. It is pre-installed your given Ubuntu environment. For more information, click **here**. Make sure that all heap blocks were freed in your code and no leaks are possible.

First, compile your code with:

```
g++ main.cpp -o main -Wall -Werror
```

which will give you an executable called "main". Then, run your executable with Valgrind, for every given case:

```
valgrind --tool=memcheck --leak-check=full --show-leak-kinds=all
./main <test case: 1 or 2 or 3> |& tee -a valgrind_log.txt
```

which will give you a log file called valgrind_log.txt. Note that in every call you will append your results to the file.

Warning: Include your created log file in your submission.

3.2 Calico

Calico is a testing tool to check whether the program gives correct output for a given input. It is preinstalled in your given Ubuntu environment. For more information, click **here**. Make sure that your code passes all the given test cases.

Run below command to test your code with given Calico file:

```
calico test.yaml |& tee calico_log.txt
```

Warning: Include your created log file in your submission.