

SCS214: Data Structures

Assignment-4: BST/Heap Trees

Instructions

- 1- Assignment should be teams of two.
- 2- Deadline of submission is **Thursday, May 11th at 11:55 pm**
- 3- Submission will be on Google classroom.
- 4- Your submission should include a single **zip file only**, named by labGroup_id1_id2.zip (ex.: NCS2_20380022_222333812.zip).
The zip file will contain **3 cpp files**: BST.h, heap.h, and labGroup_id1_id2.cpp.
(note that BST.h will have both declaration & implementation of the BST & Node classes, while heap.h will have the declaration & implementation of the heap class).
Do not use any other type of compressed files except for zip.
- 5- No late submission is allowed.
- 6- No submission through e-mails.
- 7- No exe file submission.
- 8- **In case of Cheating you will get a negative grade whether you give the code to someone, take the code from someone/internet, or even send it to someone for any reason.**
- 9- You have to write clean code and follow a good coding style including choosing meaningful variable names.
- 10- In case of wrong submission, wrong file extension/type, missing files, plagiarism, extra submitted files, compressed files, wrong naming, the assignment will not be accepted and no correction for these mistakes is allowed and you will lose your grade.

Task 1: FCAI Task Manager (BST)

We need to make a “Task Manager” application to manage users’ daily tasks where users can add a task, remove a task, search a task, display tasks based on their duration.

Each Task is represented as follows:

- Task description (e.g. study data structure 😊)
- Task duration by minutes (e.g. 60 minutes)
- Task category (e.g. educational)



You will build a class for representing “task object” and another class for representing binary search tree. The BST will store the tasks according to their **duration**, and will include the following functions:

1. Insert a task (input: description, duration, and category) where equal values of duration are added as left child.
2. Display all (in-order)
3. Search for a task (input: duration minutes, search for all tasks with the exact input duration)

SCS214: Data Structures

Assignment-4: BST/Heap Trees

4. Remove a task (input: duration minutes, remove all tasks with the exact input duration)
5. Display more than (input: duration minutes, display all tasks having $>$ or $=$ input duration)
6. Display less than (input: duration minutes, display all tasks having $<$ or $=$ input duration)

Task 2: FCAI Task Manager (Heap)

Implement Min-heap tree to store tasks objects based on task duration. The heap tree will be used when user finishes a task, it will be marked as completed and removed from the binary tree to be sorted in a heap tree. Provide the following two functions:

1. Mark task as completed (remove the task from BST and store it in min-heap)
2. Display all completed tasks based on duration ascendingly (display the completed tasks from the least task's duration) and the number of completed task per category.

<<Assignment Input/Output>>

Input File (Part 1)

```
10
Study data structure
60
Educational
Go the gym
30
Health
Watch a podcast
120
Self development
Prepare a Meal
30
Food
Study software engineering
60
Educational
Reading a book
20
Self development
```

Input File (Part 2)

```
Tidy room
15
Other
Shopping
20
Other
Go to Library
30
Other
Make DS Assignment
100
Educational
```

In the input File, first line is number of tasks then three line per task, i.e. description, time, and category. In main function, you will create an object of task and insert one by one into binary search tree.

SCS214: Data Structures

Assignment-4: BST/Heap Trees

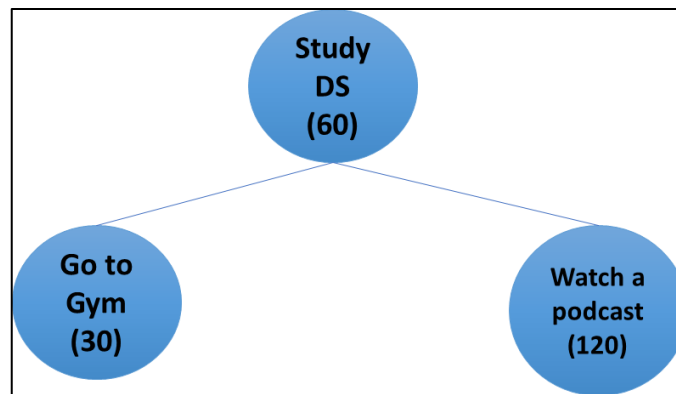


Figure 1: Example of BST after storing first 3 tasks

Then, you will show the following menu:

1. Insert a task (using BST Class)
2. Display all (using BST Class)
3. Search for a task (using BST Class)
4. Remove a task (using BST Class)
5. Display more than (using BST Class)
6. Display less than (using BST Class)
7. Mark a task as completed by task duration and description (using heap Class)
8. Display all completed tasks and the number of tasks completed per category (using heap Class)

Note that in option 7, the user might enter only part of the description and can be upper or lower case, so your search should not be case-sensitive and should match substrings.

Example:

Enter number of option: 1

Enter tasks description: Write a journal

Enter duration: 5

Enter Category: Other

The task is added.

Enter number of option: 2

//11 tasks are printed

[Write a journal, 5, Other]

[Tidy Room, 15, Other]

..

..

[Watch a podcast, 120, Self development]

Enter number of option: 3

Enter the duration: 30

//3 tasks are found

[Go the gym, 30, Health]

[Prepare a Meal, 30, Food]

[Go to Library, 30, Other]

Enter number of option: 4

Enter the duration: 30

//3 tasks are removed

Enter number of option: 5

Enter the duration more than: 100

//2 tasks are found

[Make DS Assignment, 100, Educational]

[Watch a podcast, 120, Self development]

SCS214: Data Structures

Assignment-4: BST/Heap Trees

Enter number of option: 6

Enter the duration less than: 30

//4 tasks are found

[Write a journal, 5, Other]

[Tidy room, 15, Other]

[Reading a book, 20, Self development]

[Shopping, 20, Other]

Enter number of option: 7

Task duration: 20

Task description: reading

[Reading a book, 20, Self development]

This task is removed

//choose to complete other 3 tasks

(duration = 120,100,15)

Enter number of option: 8

//min heap tree is printed

[Tidy room, 15, Other]

[Reading a book, 20, Self development]

[Make DS Assignment, 100, Educational]

[Watch a podcast, 120, Self development]

//Category report

Educational = 1

Health = 0

Self development = 2

Food = 0

Other = 1

BST and Heap Grading Criteria

Reading file into BST	10
Insert BST	10
Remove BST	10
Search BST	10
Display (all, more, less)	30
Insert min Heap	5
Display from Heap + category report	15
Main Scenario	10
Total	100