

COL106 ASSIGNMENT_5

Name: Dipen Kumar

Entry Number: 2018CS50098

Group Number: 3

All the previous data structures used in Assignment4 is used here. Their complexity were explain earlier in README_4. Also the complexity of all previous method of Scheduler were explained. Complexity of new Queries are mentioned below-

- `ArrayList<JobReport_> timed_report(String[] cmd)`
 - `NEW_PROJECT <PROJECT> <T1> <T2>`

Here I had created a field of `ArrayList<Job>` in Project class and the moment when any Job was created I added that job at last in ArrayList lying in its corresponding Project. In order to print the ArrayList of Jobs of project `<PROJECT>` between time `<T1>` to `<T2>` in order of complete first then according to arrival time, I first iterated through `printc` which was array storing completed Jobs and took those jobs satisfying above things. Then I printed all jobs in the array list of that project class verifying that it is between T1 to T2 also it is not completed. Complexity: $n+m$ where n is number of elements in arraylist containing job of project and m is the number in completed job array list.
 - `NEW_USER <USER> <T1> <T2>`

Similar to above I have stored array list of Jobs of having user `<USER>` in class `<USER>`. I first iterated through the `printc` in order to find the jobs of user `<USER>` between `<T1>` to `<T2>`. This ensure that list will have first completed comes first. Then I returned all those jobs in array list such that it is not executed and lies between `<T1>` to `<T2>`.

Complexity: $n+m$ where n is number of elements in arraylist of job in user and m is the number in completed job array list.

- NEW_PROJECTUSER <PROJECT> <USER> <T1> <T2>

Again same go through printc and return all those jobs of that user and project between time <T1> to <T2>. Then chose the array list of job stored in user class or project class which has smallest size and then iterate over it to return those between <T1> to <T2> of project and user and not completed.

Complexity: $n+m$ where n is number of elements in arraylist of job in user or project depending upon which is minimum and m is the number in completed job array list.

- PRIORITY <PRIORITY>

I had created an maxheap of project which return me project with highest priority when done extractMax. In case of conflict when two will have same priority then arrival time of project is taken care of. When I get project I check for its priority and if it is greater than given value then return all jobs in the array list of that project.

Complexity: $n+m\log(m)$ where n is the total number of jobs in those project which are of priority greater than asked for. m is the number of project in maxheap.

- ArrayList<UserReport_>timed_top_consumer(int top)

Here I had stored User when they arrive in a MaxHeap of User and when any job is executed I increase the value of the consumed in that user to which this job belong to and bubble it up in MaxHeap till it gets to its right position in maxheap. When this method is called I extract the value in maxheap of user and return till it says and in case I have low elements till its size support.

Complexity: $n \log(n)$ where n is the number of user.

- `void timed_flush(int waittime)`

This is made by iterating over maxheap of job and those who were waiting for longer or equal to given time I increased its priority by 9999 and bubbled it up. I counted the number of bubble and then extracted max that many times if it allows execution than execute it else put in not ready queue. This ensure that jobs with higher priority will come first and if two have same priority then on the basis of first in first out.

Complexity: $n \log(n)$ where n is the number of jobs in max heap.

- `void timed_handle_user(String name)`
- `void timed_handle_job(String[] cmd)`
- `void timed_handle_project(String[] cmd)`
- `void timed_run_to_completion()`

All these were implemented in same way as in assignment 4 but with only difference here is no print statement is used.

Status of Work: Completed.....