COM2031 Advanced Algorithms, Autumn Semester 2019

Lab 5: Interval Scheduling

Purpose of the lab

Your task is to implement the Interval Scheduling algorithm presented in Lectures

Interval Scheduling

Implement the Interval Scheduling algorithm that was discussed in the lecture.

In the lecture we discussed the Java implementation of the *Interval Partitioning* algorithm. Studying that implementation can give you some ideas of how to proceed in this lab.

Create a class `IntervalScheduling` that will contain all the following code:

1. Create a data type called Interval (ie a Class in Java or a struct in C++) that represents an Interval.
   (a) What fields does this Interval need to have, ie what is the information about an Interval that the algorithm needs in order to function?
   (b) For convenience, it makes sense to also add a field String name in addition to the fields that are directly required by the algorithm.
   (c) Auto-generate a constructor using Fields (see Source menu in Eclipse).
   (d) Auto-generate a toString method for easy printing

2. Add a method schedule to IntervalScheduling that implements the algorithm.
   (a) It shall take an array of Intervals and produce a maximal list of intervals that can be scheduled without overlapping (ie this is what the algorithm does).
   (b) While implementing the same algorithm, its implementation might eg make use of different data structures. It might eg return a new List with only the scheduled intervals, or it might reorder the Intervals in the array given to it in its method argument, tag those that are selected and return the number of jobs scheduled.
   (c) The method shall not print anything (expect possibly debugging message during development / debugging). (Ie: it shall not have any side effects.)
   (d) Remember from the slides the algorithm has 2 steps:
      1. sorting the jobs in ascending order of finish times.
      2. Then picking the next job that is compatible with the previously scheduled jobs.
   (e) For the first step use an appropriate method from the java library Arrays class. However you will have to implement your own Comparator.
   (f) During the 2nd step, think about how to keep track of the information about the already scheduled jobs that is required by the algorithm to decide which job is the next compatible one.

3. Implement a main method that generates some test data (ie the intervals on Slide 4 of Slide Set 02) and tests the algorithm on this. This method shall also print out the selected jobs in some form.