Currently, a government attorney schedules his team of attorneys to cases by hand with the help of spreadsheets. The goal of this assignment is to help him automate the scheduling process.

1. Use Java (C or C++) to implement:

   (a) Scheduling with Least Constraining attorney in branch/attorney ordering—the `main` method is in `LeastContstraining.java`

   (b) Scheduling with Least Workload (in days) attorney in branch/attorney ordering—the `main` method is in `LeastWorkload.java`

   (c) Extra Credit (30 points): Complete/partial scheduling (partial scheduling maximizes total scheduled case length (days)); output (to the screen) cases that are not scheduled if a complete schedule cannot be found—the `main` method is in `LeastConstrainingPartial.java`

   (d) Extra Credit (30 points): Complete/partial scheduling using beam search (partial scheduling maximizes total scheduled case length (days)); output (to the screen) cases that are not scheduled if a complete schedule cannot be found—the `main` method is in `BeamSearch.java` [discuss at the beginning of `BeamSearch.java` how you can maximize the total scheduled case length (days)]

2. Input:

   (a) judge/case/attorney file

3. Output:

   (a) screen:

      i. Number of tree nodes considered
      ii. CPU time used (in seconds)
      iii. Was a complete schedule found? If so, output the following:
          A. Number of attorneys who are not scheduled with any cases
          B. Standard deviation, minimum, and maximum number of scheduled cases for an attorney
          C. Standard deviation, minimum, and maximum number of scheduled days for an attorney

   (b) Schedule file: row=attorney, column=day, cell=caseID:judge / y / n (y=available, n=not available)

4. Provide a report (pdf):

   (a) For each toy input file (1 thru 4):

      i. draw tables for the case schedule and attorney availability (similar to those in the slides)
      ii. draw a table for the attorney schedule if a complete schedule exists; otherwise, discuss why a complete schedule does not exist and draw a table with a partial schedule.

   (b) Discuss the strength of Least Constraining and Least Workload in attorney/branch ordering. Discuss how you might combine the strength of both ordering methods.

   (c) Compare the two algorithms:

      i. output quality
      ii. time/speed
      iii. space/memory

5. Provide `readme.txt`

   (a) how to compile your programs

   (b) how to run the two algorithms

   (c) sample output of each algorithm for each input data set

6. Submit: source code, report, and `readme.txt`