Assignment 2: CS21003 Algorithms 1

Prof. Partha Pratim Chakrabarti and Palash Dey
Indian Institute of Technology, Kharagpur

Submit by 11:59 PM of February 17, 2022

1. Suppose we have n jobs with positive lengths $\ell_1, \ldots, \ell_n$ and positive deadlines $d_1, \ldots, d_n$. In a schedule σ (which is a permutation of $\{1, \ldots, n\}$), let $C_j(\sigma)$ be the finish time of job j in the schedule σ. The lateness $\lambda_j(\sigma)$ of a job j in the schedule σ is defined as $\max\{0, C_j(\sigma) - d_j\}$. Which of the following greedy algorithms outputs a schedule which minimizes total lateness $\sum_{i=1}^{n} \lambda_j(\sigma)$.

   (a) Schedule the jobs in non-decreasing order of deadline d_j.

   (b) Schedule the jobs in non-decreasing order of lengths ℓ_j.

   (c) Schedule the jobs in non-decreasing order of $d_j \ell_j$.

   (d) None of the above.

[10 Marks]

2. There are n people. A parent-child relationship is stored pairwise in an $n \times n$ matrix A where the $A[i, j]$ is marked 1 is person i is a parent of person j, otherwise it is 0. $A[i, i] = 0$ and there are no directed cycles formed through the sequence of parent-child dependencies. We are interested to efficiently find the following connections between all people:

   (a) [Ancestor] Whether person i is an ancestor of a person j, that is, connected through a sequence of parent-child relationships. A parent is also an ancestor of the child. We wish to produce a matrix B as an answer where $B[i, j]$ is marked 1 if person i is an ancestor of person j, otherwise it is 0.

   (b) [Relative] A person is a relative if they have either a common ancestor or a common descendant or both. A person is her / his own relative by definition. We are interested to find this connection between all persons. We wish to produce a matrix C as an answer where $C[i, j]$ is marked 1 if person i is a relative of person j, otherwise it is 0.

   You are to do the following for each part (a) and (b)

   i. Develop a recursive definition for the problem.
   ii. Show its working on an example of 7 persons.
   iii. Prove its correctness.
   iv. Analyze its time complexity at this initial stage.
   v. Identify scope identical sub-problems, memoization, pruning.
   vi. Define the Data Structures.
   vii. Develop the final algorithm .
   viii. Analyze the time and space complexity of the final algorithm.
   ix. Show its working on the final algorithm using the same example of 7 persons.

[10 Marks]