Tutorial 3: CS21003 Algorithms I

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

February 6, 20201

1. For the activity selection problem, find which of the following greedy strategies always output an optimal solution [Erickson, 2019].
   (a) Choose the course x that ends last, discard classes that conflict with x, and recurse.
   (b) Choose the course x that starts last, discard all classes that conflict with x, and recurse.
   (c) If no classes conflict, choose them all. Otherwise, discard the course with longest duration and recurse.
   (d) If no classes conflict, choose them all. Otherwise, discard a course that conflicts with the most other courses and recurse.
   (e) If any course x completely contains another course, discard x and recurse. Otherwise, choose the course y that ends last, discard all classes that conflict with y, and recurse.
   (f) Let x be the class with the earliest start time, and let y be the class with the second earliest start time.
      - ▷ If x and y are disjoint, choose x and recurse on everything but x.
      - ▷ If x completely contains y, discard x and recurse.
      - ▷ Otherwise, discard y and recurse.

2. Suppose there are n lectures with start and end times $S[1, \ldots, n]$ and $F[1, \ldots, n]$. Obviously, if two lectures overlap, then both of them cannot be conducted in a single hall. If two lectures do not overlap, then we are allowed to conduct them in the same hall. Design a greedy algorithm to compute the minimum number of lecture halls needed to conduct these n lectures.

3. Let X be a set of n intervals on the real line. We say that a subset of intervals $Y \subseteq X$ covers X if the union of all intervals in Y is equal to the union of all intervals in X. The size of a cover is just the number of intervals.

   Describe and analyze an efficient algorithm to compute the smallest cover of X. Assume that your input consists of two arrays $L[1 \ldots n]$ and $R[1 \ldots n]$, representing the left and right endpoints of the intervals in X. If you use a greedy algorithm, you must prove that it is correct [Erickson, 2019]. For simplicity, you may assume that the intersection of any two intervals contains either no or infinitely many real numbers.

References

[Erickson, 2019] Erickson, J. (2019). *Algorithms*.