CS 180: Homework 2

Due 4/24 at 11:59 PM

For all problems requiring an algorithm, please also provide a corresponding proof and time complexity analysis.

1. Exercise 3 on Page 107. (Extend topological sorting algorithm to return a DAG or a cycle)
2. Exercise 4 on Page 107. (Butterflies)
3. Exercise 9 on page 110. (Delete a node to destroy all paths)
4. Exercise 11 on page 111. (Networked computers)
5. Exercise 12 on page 112. (Birth & Death of people in village)
    a. Modification: in addition to the two forms of facts in the original question, a third form is possible. Changes of the question prompt are shown in bold below. Each fact has one of the following three forms:
        i. For some i and j, person Pi died before person Pj was born; or
        ii. For some i and j, person Pi died before person Pj died; or
        iii. For some i and j, the life spans of Pi and Pj overlapped at least partially.
6. Given an array arr of size N , the task is to find the minimum number of jumps to reach the last index of the array starting from index 0. In one jump, you can jump from current index i to index (i + 1) or index (i − 1); or you can move from index i to index j (i/= j) if arr[i] is equal to arr[j] + 2.
    Note: You can not jump outside of the array at any time.
    Examples:
    Input: arr = {100, −23, −23, 404, 98, 23, 23, 23, 3, 402}
    Output: 3
    Explanation: Valid jump indices are 0 → 4 → 3 → 9.