Lab 4: Decrease and conquer algorithms

Q1. Topological sorting.

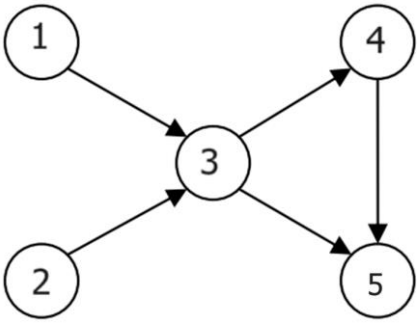
Problem statement: Apply the Decrease-and-conquer approach, decrease by constant, to find a topological order from a directed acyclic graph $G = (V, E)$, i.e., a linear order of vertices such that the edges will all point in one direction.

Input: Read the input data, which is an adjacency list of the graph $G = (V, E)$, from a text file whose format is as follows

- 1st line: a positive integer n to indicate the number of vertices in the set of vertices V
- Each of the following lines presents the list of vertices arrived from a vertex, whose index ranges from 1 to n.
  - A non-negative integer to indicate the number of vertices arrived from vertex i
  - Subsequent non-negative integers are indices of the vertices adjacent to vertex i.
  - Please mind the direction of the edges. Two consecutive numbers are separated by a space.

Output: Print to the console a topological order.

Example of input and output:

Visualization	Input text file	Output to console
	5 1 3 1 3 2 4 5 1 5 0	2 1 3 4 5

Note that you must implement the algorithm given in the lecture.

Q2. Josephus problem.

Problem statement: Let n people numbered 1 to n stand on a circle. Starting the count with person number 1, we eliminate every second person until only one survivor is left. The problem is to determine the survivor's number $J(n)$. Apply the Decrease-and-conquer approach, decrease by constant factor, to solve the problem.

Input: The user inputs the number of people n ($n > 2$)

Output: Print to the console the survivor's number.

Example of input and output:

Input from user keyboard	Output to console
8	1

Note that you must implement the algorithm given in the lecture.

Q3. Selection problem.

Problem statement: Apply the Decrease-and-conquer approach, variable size decrease, to find the k^{th} smallest number in an array $S = \{s_1, s_2, \dots, s_n\}$. Assume that the array values are in ascending order.

Input: Read the input data from a text file whose format is as follows

- 1st line: a positive integer n to indicate the size of the input array
- 2nd line: n integers, two consecutive numbers are separated by a single space " "
- 3rd line: a positive integer k to indicate the k^{th} smallest number ($1 \leq k \leq n$)

Output: Print the the k^{th} smallest number to the console.

Example of input and output:

Input text file	Output to console
8 1 2 3 4 5 6 7 8 5	5

Note that you must implement the algorithm given in the lecture.

Regulations for completing the lab work

- Each question must be implemented as an independent program in a single C++ file (of format .cpp).
- The program must receive input and return output as specified Submissions with wrong regulation will result in a "0" (zero).
- Plagiarism and Cheating will result in a "0" (zero) for the entire course.
- Contact:Here.