COMP 482 Project 7

BACKGROUND: Let's get greedy. With a game! I am going to give you an array and at each index there will be a positive integer. From index i you can walk forward in the array up to the maximum number shown in the array (for example: if arr[0] = 3 then you can move forward 1 step, 2 steps, or 3 steps, from that current position. The goal of this game is simply to determine if you can reach the last index in the array.

OBJECTIVE: Create your own greedy algorithm that runs in O(n) time that can determine if you can reach the final index in the array. Pay close attention to edge cases as I will try to break your algorithm! Remember a greedy algorithm is an algorithm that makes the best decision you can make at this current step. Do not do this with dynamic programming or you will lose credit! Also any algorithm that takes longer than O(n) time will also lose points.

Input Format: The input file will be called input7.txt and be in the same directory as the java and class files. The format of input7.txt will be a standard text file containing whitespace (spaces/tabs/newlines) separated integers. Please note to put a single space between numbers to act as a delimiter (see examples below)!

Output: simply print "yes" if you can make it or "no" if you can't make it

EXAMPLE #1: Given an input7.txt file of:
1 0 1

Then the output would be:
no
from index 0 you can walk to index 1 but at arr[1] you can't go forward at all since it has a 0.

EXAMPLE #2: Given an input7.txt file of:
2 1 1 3 0 0 4

Then the output would be:
yes
from index 0 you can walk to index 2, then walk forward 1 to index 3, then walk 3 forward to index 6 (which is the end of the array. We can ignore the final value in this array)

EXAMPLE #3: Given an input7.txt file of:
2 15 1 0 0 0 0 3
Then the output would be:
yes
from index 0 only walk forward 1 step (instead of 2) then you can walk forward 6 steps to the final spot in the array. (Note you don't have to use all the steps shown at each index position. If we used 2 at index 0 then it would be impossible. If we used 15 at index 1 we would have gone out of bounds.)

Project information and Project Submissions:
- Projects will be done **ONLY** in **Java** (No other languages will be accepted)
- Students should begin to work on projects when the project specifications are released.
- Projects will be released as early as possible to students, and you are encouraged to complete the projects as early as possible. Even if a topic has yet to be covered, if students have taken the time to learn the material beforehand, feel free to attempt projects early and submit them early.
- You will be able to submit your project as many times as you wish until the deadline given. (**For a regrade you must submit the project at least 7 days prior to the deadline otherwise there will be no regrade**)
- Late projects will not be accepted.

Projects must be submitted as follows:
- You must submit your project to Canvas **ONLY.** Email submissions will not be accepted.
- The file must be submitted in a ".zip" format
- The ".zip" file should be named with your First and Last name with the project number at the end (Example: DinoBiel1.zip)

- If working in visual studio please do not zip the entire project. Only zip the ".java" file (doing this will cost you points!)
- you must use the DEFAULT package, if not sure how to do this, ask the professor.
- Failure to comply with the rules above will result in a major loss of points on the project!

Grading Rubric: When grading your projects I will assign grades based on the following criteria:
- Does the project work according to the specification including reasonable time complexity? – **50%**
- Does the project utilize the concepts requested in the specification? – **30%**
- Is the code provided in the project well formatted? – **10%**
- Does the code contain sufficient and useful comments to explain sections of the code? – **10%**