Algorithms Laboratory (CS29203)
Assignment 1: Running time of algorithms
Department of CSE, IIT Kharagpur

8th August 2024

Question-1

The famous film director Shyam Benegal is making a new movie. There has been several shots of the movie that has to be combined to complete the production. The shots are numbered as $1, 2, 3, \cdots, n$. The production will be complete only if the shots are placed in proper sequence while editing. An improper sequence of shot placement will ruin everything and the final version of the movie will make no sense. That is why Shyam Benegal is seeking help from second year IIT Kgp CSE students to make a proper sequence and successfully complete the movie editing process before releasing it.

Let the shot sequence is stored in an array $\mathcal{S}$ of size n. Each shot is denoted by an integer number in the range from 1 to n, and each shot appears only once in $\mathcal{S}$. Let us consider any three different shots s_1, s_2, s_3 where $1 \leq s_1 < s_2 < s_3 \leq n$.

Now an invalid sequence of shots is described by the one where s_1, s_2, s_3 are placed in "largest-smallest-intermediate" order, *i.e.* s_1 is placed after s_3, and s_2 is placed after s_1. Any other arrangement of shots is considered as valid, e.g. "smallest-largest-intermediate" (s_1 followed by s_3 followed by s_2), etc. For example, consider 10 frames with a sequence of $3, 4, 5, 2, 6, 7, 9, 8, 1, 10$. This is a valid sequence since it obeys the mentioned rule. However the sequence $3, 4, 5, 1, 6, 7, 9, 8, 2, 10$ is not a valid sequence because of the placement of frames $1, 2, 5$ is in "largest-smallest-intermediate" order.

In the following settings, you have to check whether a sequence of frames is valid or not.

(a) Brute force solution (10 points): Given the frame sequence, check for all combinations of s_1, s_2, s_3 and verify whether the rules are maintained or not. Note that there are $\binom{n}{3}$ frame combinations in total. Also since $\mathcal{S}$ is not sorted, you have to make linear search to locate the frame positions. Hence the running time of the algorithm will be $O(n^4)$.

(b) Slightly better brute force solution (15 points): Let us consider three frame sequences s_1, s_2, s_3. In the previous part we were running the search on s_1, s_2, s_3, but this time we will run the search on their indices in the array. For each choice of indices that satisfies $0 \leq i < j < k \leq n-1$, take $s_3 = \mathcal{S}[i], s_1 = \mathcal{S}[j]$ and $s_2 = \mathcal{S}[k]$. Then check whether $s_1 < s_2 < s_3$ or not. Since every frame appears at some position in $\mathcal{S}$, this exhausts all frame combinations. Still you can improve the solution by a factor of n with a running time of $O(n^3)$. (Feel free to use any other idea to obtain a $O(n^3)$ solution)

(c) More better solution (20 points): For each i, set $s_3 = \mathcal{S}[i]$. Now examine the subsequence of $\mathcal{S}[i+1, n-1]$ consisting of numbers $< s_3$. Now $\mathcal{S}$ is valid if and only if for each i, the subsequence is strictly decreasing. So for each i, you need to make a single pass through rest of the sequence, and eventually improve the solution by a factor of n. So the running time of the algorithm will be $O(n^2)$. (Feel free to use any other idea to obtain a $O(n^2)$ solution)

(d) Linear time solution (30 points): Design a $O(n)$ time algorithm for the same problem.

Example 1:

```
(Input) Enter the number of frames (n) = 25
        Enter the frame sequence: 3 5 8 7 6 12 14 16 15 18 21 25 24 23 22 20 19 17 13 11 10 9 4 2 1

(Output) Method(a): Valid sequence
         Method(b): Valid sequence
         Method(c): Valid sequence
         Method(d): Valid sequence
```

Example 2:

```
(Input) Enter the number of frames (n) = 25
        Enter the frame sequence: 8 19 9 4 25 5 7 2 6 10 11 12 13 14 15 16 17 21 18 3 20 23 24 22 1

(Output) Method(a): Invalid sequence
         Method(b): Invalid sequence
         Method(c): Invalid sequence
         Method(d): Invalid sequence
```

Question-2

(25 points) As an extension of the previous question, consider a given array $\mathcal{S}$ of frames. You have to find the frame index before which all the frame numbers are smaller and after which all the frame numbers are greater. Note that there are no repeated frames. This problem can be solved in $O(n^2)$ in Brute force method. Your task is to solve this problem in **linear time** by using only **one auxiliary array**.

Example 1:

```
(Input) Enter the number of frames (n) = 8
        Enter the frame sequence: 4  2  3  5  1  6  9  7

(Output) Required frame index is 5
```