Recitation 2 09/16

Plan:
1. Correctness & Complexity of \( O(n \log n) \) Algorithm
2. \( O(n) \)
3. Another \( O(n) \) Algorithm
4. Counterexample I
5. Counterexample II

Algorithm 1: \( O(n \log n) \)

(slightly different from the one in lecture)

\[
A = \begin{bmatrix}
3 & 5 & 3 & 5 & 7 \\
7 & 4 & 7 & 2 & 8 \\
15 & 3 & 3 & 3 & 1 \\
11 & 1 & 7 & 12 & 2 \\
8 & 11 & 1 & 5 & 2 \\
\end{bmatrix}
\]

1. Consider projection \( A' \) where each entry is the maximum of the corresponding column of \( A \)

2. Find the 1D-peak in \( A' \) using \( O(\log n) \) Alg

3. Find its occurrence in the corresponding column of \( A \)

It must be a 2D-peak. Why?
Computing $A'$ explicitly: $\Theta(n^2)$ time is too much.
Computing a single entry of $A'$: $\Theta(n)$ time.

1D-peak locator only looks at $O(n \log n)$ entries of $A'$.
It suffices to compute them! $O(n \log n)$

Total computation time.

Algorithm 2: $O(n)$ from the lecture.

Want to prove by induction a stronger statement:
The algorithm finds a 2D-peak that is $\geq$ anything on the boundary.
Clearly works for small matrices.

For large ones:

The peak must be greater than anything around so is a 2D-peak in the entire matrix.

Time complexity $T(n, m)$:

$$T(n, m) = O(n + m) + T\left(\frac{n}{2}, \frac{m}{2}\right)$$

$$= O(n + m) + O\left(\frac{n + m}{2}\right) + O\left(\frac{n + m}{4}\right) + \ldots + O(1)$$

Geometric sequence

$$= O(n + m)$$
Algorithm 3: $O(n)$ In Expectation

Recall: Greedily ascending can take $\Omega(n^2)$ time

Is there a way to fix it?

Idea 1: Start from a random location

$\rightarrow$ Can still take $\Omega(n^2)$ time most of the time

Idea 2: Pick $n$ random locations

- Find their maximum
- Greedily ascend from there

Intuition: If there is a very long $\Omega(n^2)$ increasing path, very likely to draw a location within the last $O(n)$ locations
ANALYSIS:

\[ \rightarrow \text{Consider the sorted sequence of all the numbers:} \]
\[ a_1 \leq a_2 \leq \ldots \leq a_{n^2-2} \leq a_{n^2-1} \leq a_{n^2} \]

\[ \rightarrow \text{If the location containing } a_i \text{ is selected, then can ascend for at most } n^2 - i \text{ steps.} \]

\[ \rightarrow \text{Probability ascends for } \gamma s \text{ steps:} \]
\[ \leq \text{No number in } a_{n^2-s+1} \ldots a_{n^2} \text{ selected} \]
\[ \leq \left( \frac{n^2 - s}{n^2} \right)^n \leq \left( 1 - \frac{s}{n^2} \right)^n \leq e^{-\frac{s \cdot n}{n^2}} = e^{-s/n} \]

\[ \rightarrow \text{Expected number of steps:} \]
\[ \leq \sum_{i=1}^{n} i \cdot \Pr[\text{walks for } (i-1)n+1 \ldots \text{in steps}] \]
\[ \leq n + \sum_{i=2}^{n} i \cdot \Pr[\text{walks for } \geq (i-1)n \text{ steps}] \]
\[ \leq n + n \cdot \sum_{i=2}^{n} i \cdot e^{-(i-1)} = O(n) \]
\[ = O(1) \]
HOW ABOUT 3 DIMENSIONS?

CAN GENERALIZE THE ALGORITHMS AS FOLLOWS

ALGORITHM 1: \(O(n^2 \log n)\) \(\leftrightarrow\) PROJECT INTO 1D

ALGORITHM 2: \(O(n^2)\) \(\leftrightarrow\) PROJECT INTO 2D

ALGORITHM 3: \(O(n^{1.5})\) \(\leftrightarrow\) START FROM THE BEST OUT OF \(n^{1.5}\) RANDOM LOCATIONS

COUNTEREXAMPLE 1

CAN USE ANY PEAK INSTEAD OF THE HIGHEST PEAK FOR COLUMNS IN ALGORITHM 1?

NO:

THERE MAY BE ONLY ONE PEAK AND WE DON'T LEARN IN WHICH HALF IT LIES, SO WE CAN'T RECURSE
COUNTEREXAMPLE II

QUESTION: ALGORITHM 2 LOOKS AT 3 COLUMNS & 3 ROWS:

![Diagram of a 3x3 grid]

IS IT POSSIBLE TO LOOK AT JUST ONE ROW & ONE COLUMN

AND RECURSE BASED ON THEM?

ANSWER: NO. CAN RECURSE ON A SQUARE THAT HAS NO 2D-PEAK IN THE ENTIRE ARRAY, EVEN THOUGH HAS A 2D-PEAK FOR A SUB ARRAY

FIRST MAX
THE ONLY PEAK
SECOND MAX
RECURSION GOES THERE BUT THERE IS NO PEAK THERE

↑-GRADIENT

PSET 1 SHOWS HOW TO FIX THIS SOLUTION