Algorithms – CMSC 37000

Dynamic programming: The all-ones square problem

Instructor: László Babai   Ry-164   e-mail: `laci@cs.uchicago.edu`

Problem. Given an $n \times n$ array A of zeros and ones, find the maximum size of a contiguous square of all ones. (You do not need to locate such a largest all-ones square, just determine its size.) Solve this problem in *linear time.* "Linear time" means the number of steps must be O(size of the input). In the present problem, the size of the input is $O(n^2)$. Manipulating integers between 0 and n counts as one step; such manipulation includes copying, incrementing, addition and subtraction, looking up an entry in an $n \times n$ array.

Describe your solution in **pseudocode.** The solution should be *very simple,* no more than a few lines. **Elegance counts.**

Example:

$$\begin{array}{|cccccc|}
\hline
1 & 0 & 1 & 1 & 0 & 1 \\
1 & 1 & 0 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 0 \\
1 & 1 & 1 & 0 & 1 & 1 \\
\hline
\end{array}$$

In this example, the answer is 3. There are three contiguous 3×3 square subarrays of all-ones. One is indicated below by <u>underlines</u>, another is shown in a $\boxed{\text{box,}}$ the third one is indicated by *Italics.*

$$\begin{array}{|cccccc|}
\hline
1 & 0 & 1 & 1 & 0 & 1 \\
1 & 1 & 0 & \textit{1} & \textit{1} & \textit{1} \\ \cline{3-5}
1 & 0 & \multicolumn{1}{|c}{1} & \textit{1} & \multicolumn{1}{c|}{\textit{1}} & \textit{1} \\
\underline{1} & \underline{1} & \multicolumn{1}{|c}{\underline{1}} & \textit{1} & \multicolumn{1}{c|}{\textit{1}} & \textit{1} \\
\underline{1} & \underline{1} & \multicolumn{1}{|c}{\underline{1}} & 1 & \multicolumn{1}{c|}{1} & 0 \\ \cline{3-5}
\underline{1} & \underline{1} & \underline{1} & 0 & 1 & 1 \\
\hline
\end{array}$$