COS 423	Theory of Algorithms	Spring 2018

Precept 6

These problems will be solved in precept.

1. *Boolean matrix multiplication.* Given two n-by-n boolean matrices A and B, the n-by-n product $C = A \times B$ is defined by:

$$c_{ij} = \bigvee_{k=1}^{n} a_{ik} \wedge b_{kj},$$

where $\vee$ denotes boolean OR, and $\wedge$ denotes boolean AND.

   (a) Design an algorithm to multiply two n-by-n boolean matrices using $O(n^3)$ bit operations.

   (b) Why doesn't Strassen's algorithm (modified to use OR instead of addition and AND instead of multiplication) work for multiplying boolean matrices?

   (c) Design an algorithm to multiply two n-by-n boolean matrices using $O(n^{\log_2 7})$ arithmetic operations, where adding, subtracting, or multiplying two integers (of any size) is counted as one arithmetic operation.

   (d) Design an algorithm to multiply two n-by-n boolean matrices using $O(n^{\log_2 7} \log^2 n)$ bit operations.

2. *Box stacking.* Given n types of rectangular 3D boxes, where box type i has width w_i, depth d_i, and height h_i, your goal is create a stack of boxes that is as tall as possible. You may stack one box on top of another box only if the width and depth of the top box are each strictly smaller than the width and depth of the bottom box. You can rotate a box so that any two sides functions as its width and depth. You may use as many instances of each box type as desired.

   For example, if the $n = 4$ box types are: $1 \times 2 \times 3$, $4 \times 5 \times 6$, $4 \times 6 \times 7$, and $10 \times 12 \times 32$, then the maximum height is $60 = 3 + 1 + 6 + 4 + 4 + 32 + 10$.

$$\begin{array}{r}
1 \times \ \ 2 \times \ \ 3 \\
2 \times \ \ 3 \times \ \ 1 \\
4 \times \ \ 5 \times \ \ 6 \\
5 \times \ \ 6 \times \ \ 4 \\
6 \times \ \ 7 \times \ \ 4 \\
10 \times 12 \times 32 \\
12 \times 32 \times 10
\end{array}$$

   Your algorithm should take $O(n^2)$ time and use $O(n)$ space.