1. Prove item (3) of Lemma 19.1 (there are exactly \( \binom{k}{i} \) nodes at depth \( i \) at \( B_k \)).


3. Question 20.2-1 from CLRS.

4. Question 20.2-4 from CLRS.

5. Question 20.2-5 from CLRS.

6. Question 20.4-1 from CLRS.

7. (This question has more than one solution, and basically the question’s purpose is to trigger you to seek nice solutions.

Let \( P \) be an \( m \times m \) boolean matrix, and \( T \) be an \( n \times n \) boolean matrix, where \( m \) is much smaller than \( n \). Each element in these matrices is either 0 or 1. Suggest an algorithm that checks if there are values \( s_1, s_2 \) such that

\[
P[i_1, i_2] = T[i_1 + s_1, i_2 + s_2],
\]

for every pair of integers \( i_1, i_2 \) such that \( 1 \leq i_1 \leq m \) and \( 1 \leq i_2 \leq m \).

8. Given points \( p, q, r \), the triangle determined by \( pqr \) is the triangle whose vertices are \( p, q, r \). The perimeter of this triangle is the sum of lengths of its edges.

Let \( S \) be a set of \( n \) points in the plane. Suggest an algorithm whose **expected** running time is \( O(n) \), and finds the triangle determined by 3 points of \( S \), and its perimeter is no larger than the perimeter of any triangle determined by triple of points of \( S \). What is the worst case running time of this algorithm?