1. See the lecture notes for the solutions. The results will be as follows:
   a) \( x(n) \in \Theta(n^{\log_b a}), a \neq 1 \)
      \( \in \Theta(\log n), a = 1 \)
   b) \( x(n) \in \Theta(n^{\log_b a}), a > b \)
      \( \in \Theta(n), a < b \)
      \( \in \Theta(n \log n), a = b \)

2. See the lecture notes.

3. Breadth-first search algorithm is an optimal algorithm for the shortest path problem. The algorithm has the complexity \( W(n, m) \in \Theta(n + m) \). See the lecture notes.

4. We break the matrices into \( n \times n \) matrices. Let \( A \) be the \( kn \times n \) matrix and \( B \) be the \( n \times kn \) matrix. Then
   \[
   A = [A_1 A_2 ... A_k]^T, \quad B = [B_1 B_2 ... B_k]
   \]
   where \( A_i \)'s and \( B_i \)'s are \( n \times n \) sub-matrices.

   Now, we can easily find the product \( AB \), which is simply
   \[
   \begin{bmatrix}
   A_1B_1 & A_2B_2 & ... & A_kB_k \\
   A_2B_1 & ... & ... & ... \\
   ... & ... & ... & ... \\
   A_kB_1 & A_kB_k
   \end{bmatrix}
   \]

   Thus, \( AB \) can be expressed as a \( k \times k \) matrix, each entry of which is a product of two \( n \times n \) matrices. By using Strassen’s algorithm to do each of the multiplications, we get a running time of \( \Theta(k^2 n^{2.81}) \), since there are \( k^2 \) entries, and each entry requires a single multiplication.