1. (15 Punkte) Wir haben das Problem besprochen, für eine Menge von n Schlüsseln $x_1 < x_2 < \cdots < x_n$ mit vorgegebenen Zugriffsfrequenzen $f_1, \ldots, f_n$ einen optimalen binären Suchbaum zu berechnen.

   (a) Eigentlich haben wir nur gezeigt, wie die "Kosten" dieses optimalen Suchbaums in $O(n^3)$ Zeit berechnet werden können. Zeigen Sie, dass auch der optimale Baum in dieser Zeit berechnet werden kann.

   (b) Die Annahme in der Vorlesung war, dass es nur erfolgreiche Zugriffe gibt, also nur mit Schlüsseln x gesucht wird, die tatsächlich einem der x_i's gleichen. Nehmen wir nun an, das gilt nicht und es gibt auch erfolglose Zugriffe. Für jedes i mit $0 \leq i \leq n$ sei nun g_i die Zugriffsfrequenz für Schlüssel x mit $x_i < x < x_{i+1}$. Dabei gilt $x_0 = -\infty$ und $x_{n+1} = +\infty$. Die Kosten für einen erfolglosen Zugriff für so ein x in einem Baum T sind dann die Anzahl der Schlüsselvergleiche zwischen x und den auf dem Suchpfad gespeicherten Schlüsseln (also die Länge des sich in T ergebenden Suchpfades).

   Geben Sie einen Algorithmus an, der für gegebene Folgen f_i von Frequenzen von erfolgreichen Zugriffen und g_i von Frequenzen von erfolgsosen Zugriffen die Kosten des optimalen Suchbaums berechnet. Die Laufzeit des Algorithmus sollte polynomiell sein.

2. (30 Punkte) In dieser Aufgabe geht es um Matrizenmultiplikation. Der übliche Multiplikationsalgorithmus benötigt $O(p \cdot q \cdot r)$ Zeit, um das Produkt einer $p \times q$ und einer $q \times r$ Matrix zu berechnen. Wir sagen, diese Produktberechnung hat Kosten pqr. Wie man inzwischen weiß, ist das überraschenderweise asymptotisch nicht optimal, aber das soll nicht Thema dieser Aufgabe sein. Hier geht es um die effiziente Berechnung des Produkts einer Folge von Matrizen $A_1, \ldots, A_n$, also die Berechnung von $A_1 \cdot A_2 \cdot \cdots \cdot A_n$. Dabei habe jede Matrix A_i die Dimensionen $p_{i-1} \times p_i$ für ein Folge von positiven ganzen Zahlen $p_0, \ldots, p_n$. Dies stellt sicher, dass das Gesamtprodukt tatsächlich definiert ist (es hat Dimension $p_0 \times p_n$). Da Matrizenmultiplikation assoziativ ist, kann bei der Berechnung des Gesamtproduktes beliebig geklammert werden, also z.B. $(A_1 \cdot A_2) \cdot A_3$ oder $A_1 \cdot (A_2 \cdot A_3)$.

   (a) (4 Punkte) Zeigen Sie, dass es bei verschiedenen Klammerungen zu verschiedenen Gesamtkosten bei der Berechnung des Gesamtproduktes kommen kann.

   (b) (6 Punkte) Was ist der größte Unterschied der Gesamtkosten, den Sie erzielen können bei einer Gesamteingabegröße von $N = p_0p_1 + p_1p_2 + \cdots + p_{n-1}p_n$? Genauer gesagt, Sie sollen für unendlich viele N jeweils irgendwelche $p_0, \ldots, p_n$ wählen mit $\sum_{1 \leq i \leq n} p_{i-1}p_i = N$ und jeweils zwei Klammerungen wählen, sodass die zwei sich nach diesen beiden Klammerungen ergebenden Gesamtkosten möglichst unterschiedlich sind (gemessen in N).

   (c) (15 Punkte) Geben Sie einen Algorithmus an, der die bestmöglichen Gesamtkosten für die Berechnung des Gesamtproduktes berechnet. Die Laufzeit des Algorithmus sollte polynomiell in n sein.

   (d) (5 Puntke) Erweitern Sie Ihren Algorithmus, so dass er auch die entsprechende Klammerung angibt (durch den entsprechenden Ausdrucksbaum).