

Aufgabe 1:

1. Gesucht werden soll das Muster “AGGG” in der Sequenz “GGGGGGGG”. Beschreiben Sie den Ablauf des Verfahrens von Boyer und Moore
 - (a) unter ausschließlicher Verwendung der *bad character*-Heuristik und
 - (b) bei zusätzlicher Anwendung der *good suffix*-Heuristik.
2. In der Vorlesung wurde die *bad character*-Heuristik des Boyer-Moore-Algorithmus vorgestellt, die zum Einsatz kommt, wenn das erste verglichene Zeichen (das letzte im Suchmuster) einen Mismatch verursacht.
 - (a) Implementieren Sie eine Funktion `calcLast`, die aus einem Muster $P = p_1 \dots p_n$ eine `HashMap` `last` berechnet, welche zu jedem p_i aus P den grössten Index j liefert, so dass $p_i = p_j$ gilt.

Beispiel: Für $P = \text{affe}$ ergibt sich `last` wie folgt:

last		
a	f	e
1	3	4

- (b) Implementieren Sie basierend auf `calcLast` eine Variante des Boyer-Moore-Algorithmus, die nur die *bad character*-Heuristik unterstützt.
- (c) Erweitern Sie die *bad character*-Heuristik so, dass Sie auch dann eingesetzt werden kann, wenn es bei einem anderen als dem letzten Zeichen im Muster zu einem Mismatch gekommen ist.

Hinweis: `last` wird zu einem zweidimensionalen Feld.

Aufgabe 2:

Geben Sie für den folgenden Graph an,

- in welcher Reihenfolge die Knoten bei einer Tiefensuche (DFS) besucht werden.
- in welcher Reihenfolge die Knoten bei einer Breitensuche (BFS) besucht werden.

Starten Sie dabei immer am Knoten 1 und wenn Sie in einem Schritt eine Auswahl zwischen mehreren Knoten haben, wählen Sie den mit der kleineren ID.

