

Aufgabe 2 (8 Punkte)

Betrachte das Wort $t\$ = t_1 \cdots t_{10}\$ = \text{ANNABANANA}\$$.

- Konstruiere die Burrows-Wheeler-Transformierte \hat{t} zu $t\$$.
- Gib die zugehörige LF-Funktion für \hat{t} an.
- Bestimme die Werte $C(\cdot)$ und $Occ(\cdot, \cdot)$.
- Suche nach $s = s_1 \cdots s_3 = \text{BAN}$ im FM-Index für t mit Hilfe des in der Vorlesung angegebenen Algorithmus.

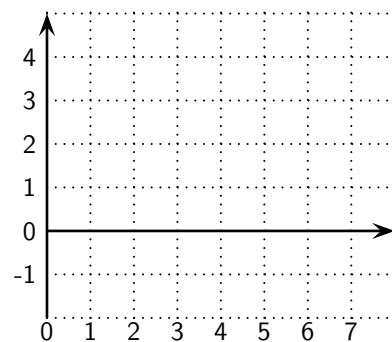
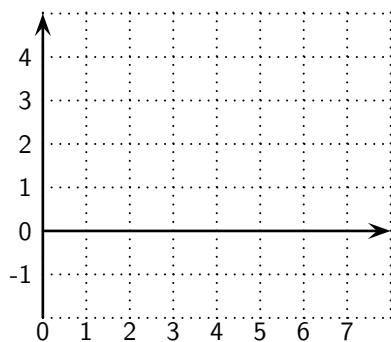
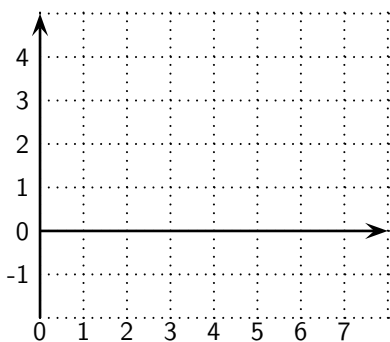
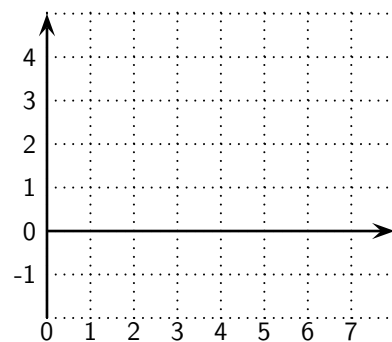
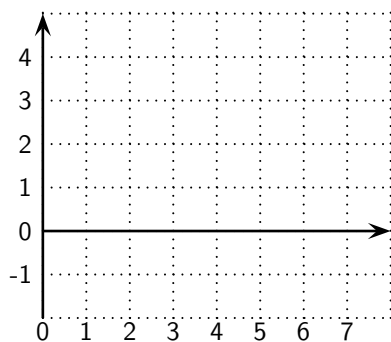
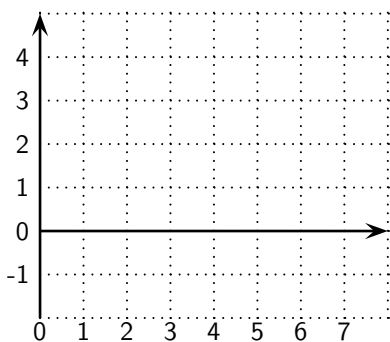
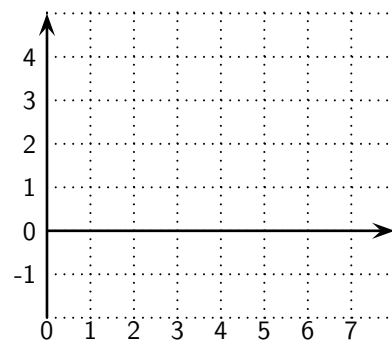
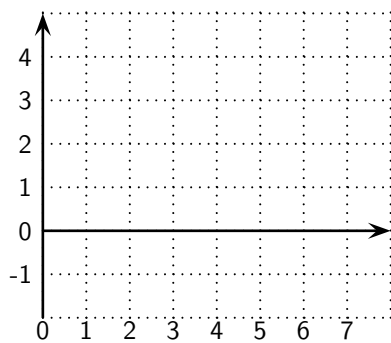
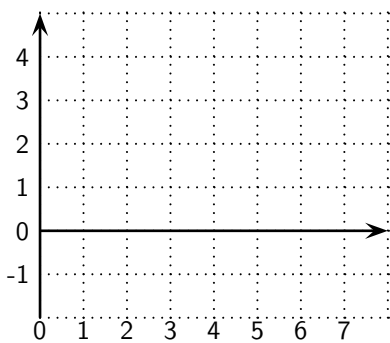
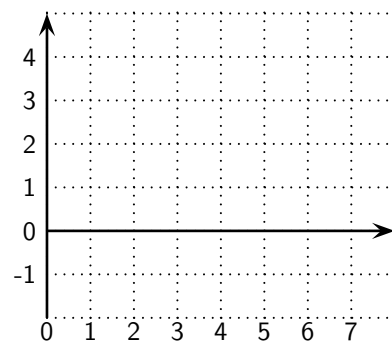
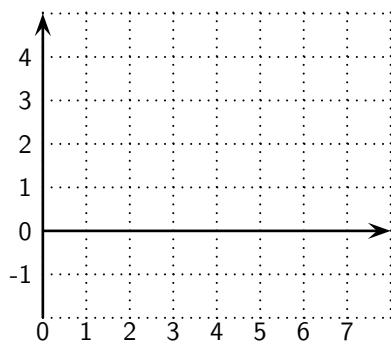
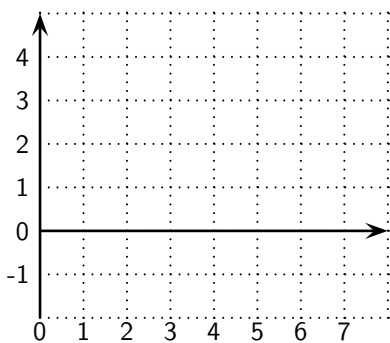
Hinweise: Für Teil a) und b) fülle die unten angegebene Tabelle korrekt aus. In dieser Aufgabe gilt: $\$ < A < B < N$.

i	$A[i]$	$t^{A[i]}$	\hat{t}_i	LF[i]
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

Aufgabe 3 (8 Punkte)

Ermittle mit dem in der Vorlesung angegebenen Linearzeit-Algorithmus für AMSS alle maximal bewerteten Teilfolgen von a und gib dabei alle Zwischenschritte sowie die jeweils angewendeten Fälle an. Gib auch an, wann welche maximal bewertete Teilfolge vom Algorithmus ausgegeben wird.

$$a = (+4, -3, +2, -1, +3, -6, +3)$$



Vorname: _____ Name: _____ Matrikelnummer: _____

Aufgabe 4 (8 Punkte)

Entwirf für das LCE-Problem einen alternativen Algorithmus basierend auf Extended Suffix-Arrays (ohne Verwendung von Bäumen), so dass Anfragen in konstanter Zeit nach einer linearen Vorverarbeitung beantwortet werden können. Versuche dabei, möglichst wenig Bestandteile des Extended Suffix-Arrays zu verwenden.

Hinweis: Sei $t \in \Sigma^n$ und sei $t' = t\$$. Für $i < j \in [1 : n]$ ist die longest common (forward) extension von i und j in t definiert als die Zahl $\ell \in \mathbb{N}_0$ mit $t'_i \cdots t'_{i+\ell-1} = t'_j \cdots t'_{j+\ell-1}$ und $t'_{i+\ell} \neq t'_{j+\ell}$ (in Zeichen $\text{lce}(i, j) = \ell$).

Vorname: _____ Name: _____ Matrikelnummer: _____

Aufgabe 5 (8 Punkte)

Sei Σ ein Alphabet und $t \in \Sigma^n$. Konstruiere einen Algorithmus mit Laufzeit $O(n)$, der alle Wörter $w \in \Sigma^*$ findet, die in t mindestens zweimal und höchstens viermal auftreten, wobei kein echtes Präfix von w auch maximal viermal in t auftritt.

Zeige die Korrektheit des Algorithmus und analysiere dessen Laufzeit.

Hinweis: Für $w \in \Sigma^*$ ist $u \in \Sigma^*$ ein *echtes Präfix*, wenn $w = uv$ mit $v \in \Sigma^+$ ist.