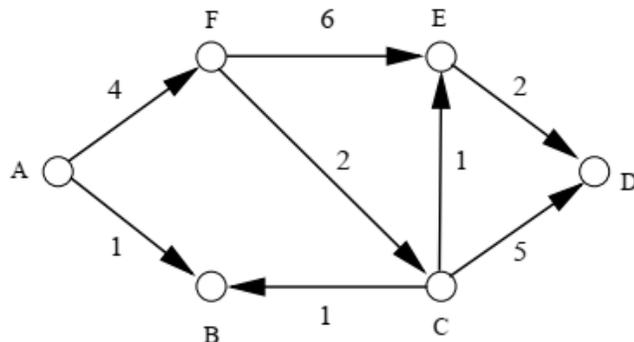


16. Übung

Einführung in die Bioinformatik I, 2. Teil
Sommersemester 2021

Aufgabe 1 (7 Punkte): Gegeben ist der folgende gerichtete, gewichtete Graph:



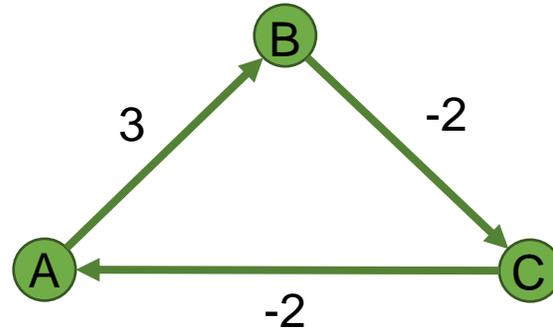
Verwenden Sie Dijkstras Algorithmus, um einen kürzesten Pfad von A nach D zu finden. Beschreiben Sie den Zustand der Datenstrukturen für die Schritte des Algorithmus.

Schritt	Pfade	Queue	Distanzen					
Init		(A, 0)	A	B	C	D	E	F
			0	∞	∞	∞	∞	∞
1	A nach F; A nach B;	(B, 1) (F, 4)	0	1	∞	∞	∞	4
2	Von B keine ausgehenden Kanten;	(F, 4)	0	1	∞	∞	∞	4
3	F nach C; F nach E;	(C, 6) (E, 10)	0	1	6	∞	10	4
4	C nach B; C nach E; C nach D;	(E, 7) (D, 11)	0	1	6	11	7	4
5	E nach D;	(D, 9)	0	1	6	9	7	4
6	Von D keine ausgehenden Kanten;	Leer \rightarrow Ende	0	1	6	9	7	4

Aufgabe 2 (3 Punkte): Wieso kann der Dijkstra-Algorithmus nicht mit negativen Kantengewichten umgehen? Begründen Sie und geben Sie ein Beispiel an, für das der Algorithmus nicht funktioniert.

Aufgabe 2 (3 Punkte): Wieso kann der Dijkstra-Algorithmus nicht mit negativen Kantengewichten umgehen? Begründen Sie und geben Sie ein Beispiel an, für das der Algorithmus nicht funktioniert.

Wenn negative Kreise auftauchen, findet der Algorithmus immer kleinere Distanz-Werte für die Knoten im Kreis, so dass der Algorithmus nicht terminiert.



Schritt	Pfade	Queue	Distanzen
Init		(A, 0)	A B C 0 ∞ ∞
1	A nach B;	(B, 3)	0 3 ∞
2	B nach C;	(C, 1)	0 3 1
3	C nach A;	(A, -1)	-1 3 1
4	A nach B;	(B, 2)	-1 2 1
5	und so weiter

Wie helfen uns Graphen das multiple Sequenzalignmentproblem (zumindest heuristisch) effizient zu lösen?

Warum können negative Kantengewichte für Alignmentgraphen wichtig sein?

Wie könnte man den Dijkstra-Algorithmus verändern, damit er doch mit negativen Kantengewichten umgehen kann?

Welche Auswirkungen hätten diese Veränderungen? Könnte er dann immer noch in jedem Fall eine optimale Lösung finden? Wenn ja/Wenn nein, wie müssten diese Fälle aussehen?