Intervall Scheduling multiple choice

Gegeben seien folgende Aufgaben
(jeweils mit Startzeit, Terminierungszeit und Wert):
A_1 (2,6,16) A_2 (1,3,5) A_3 (4,8,8) A_4 (5,6,1).

Welche Aufgaben wählt der Algorithmus aus?

Auflösung: A_2 & A_4

Welche Auswahl ergäbe das höchste Gesamtgewicht?

Auflösung: A_1

Minimieren maximaler Verzögerung demogr.

Gegeben seien drei Aufgaben A_i mit Werten $(\mathrm{Frist}_i, \mathrm{Dauer}_i)$:
$A_1 = (5, 4)$, $A_2 = (6, 8)$, $A_3 = (2, 1)$.

Welche maximale Verzögerung ergibt sich bei optimaler Anordnung?

Auflösung: Verzögerung 7 bei Anordnung A_3 A_1 A_2

Huffman-Codes Beispiel multiple choice

Gegeben folgender Text über dem Alphabet $\Sigma = \{a, b, c, d\}$:

d c d c b a b a b a

Welches ist ein Huffman-Code für diesen Text:

- (1) $a \rightarrow$ 00, $b \rightarrow$ 01, $c \rightarrow$ 10, $d \rightarrow$ 11
- (2) $a \rightarrow$ 0, $b \rightarrow$ 1, $c \rightarrow$ 00, $d \rightarrow$ 01
- (3) $a \rightarrow$ 0, $b \rightarrow$ 10, $c \rightarrow$ 110, $d \rightarrow$ 111
- (4) $a \rightarrow$ 11, $b \rightarrow$ 10, $c \rightarrow$ 01, $d \rightarrow$ 00

Auflösung: (1) & (4)

Einordnung von Quicksort multiple choice

Quicksort ist ein Vertreter

- (1) der Greedy-Algorithmen.
- (2) der Divide-and-Conquer Algorithmen.
- (3) der dynamischen Programmierung.
- (4) keiner dieser Klassen.
- (5) keine Ahnung.

Auflösung: (2) (und damit auch (3))

Dynamische Programmierung demogr.

Bei allgemeiner dynamischer Programmierung ist der unterliegende Aufrufgraph kein Baum (sondern ein DAG). Deshalb ist der benötigte Speicherplatz immer superlinear in der Länge der Eingabe.

- (1) Stimmt.
- (2) Stimmt nicht.
- (3) Keine Ahnung.

Auflösung: (2)

Bsp: $O(n)$ Platz für Fibonacci-Folge $f(n) = f(n-1) + f(n-2)$.

Gewichtetes Intervall-Scheduling (1) demogr.

Welche Laufzeit hat die dynamische Programmierung für gewichtetes Intervall-Scheduling?

- (1) $\Theta(n)$
- (2) $\Theta(n \log n)$
- (3) $\Theta(n^2)$
- (4) $\Theta(n^2 \cdot 2^n)$
- (5) $\Theta(n!)$

Auflösung: (2) $\Theta(n \log n)$

Gewichtetes Intervall-Scheduling (2) demogr.

Betrachte eine beliebige Instanz mit n Aufgaben. Sei

- g = Anzahl Aufgaben, die von Greedy ausgewählt werden.
- d = Anzahl Aufgaben, die von dyn. Program. ausgewählt werden.

Wie groß kann g/d werden?

- (1) $\Theta(1)$
- (2) $\Theta(\sqrt{n})$
- (3) $\Theta(n)$
- (4) $\Theta(n \log n)$

Auflösung: (3) $\Theta(n)$

APSP Algorithmus von Bellman-Ford demogr.

Welche Bedingungen benötigt der Algorithmus zur korrekten Ausführung?

- (1) Positive Kantengewichte
- (2) Symmetrische Kantengewichte
- (3) Stark zusammenhängender Graph
- (4) Zusammenhängender Graph
- (5) Kreisfreier Graph (DAG)
- (6) Keine dieser Bedingungen
- (7) Ich kenn nur Ford-Mustang...

Auflösung: (6)

APSP Algorithmus von Floyd (1) demogr.

Welche Laufzeit hat der Algorithmus von Floyd für einen gerichteten Graphen mit n Knoten und m Kanten?

- (1) $\Theta(n \cdot m^2)$
- (2) $\Theta(n^2 \cdot m)$
- (3) $\Theta(n^3)$
- (4) $\Theta(n^{\log_2 7} \cdot m)$
- (5) Floyd?

Auflösung: (3) $\Theta(n^3)$

APSP Algorithmus von Floyd (2) demogr.

Welche Bedingungen benötigt der Algorithmus zur korrekten Ausführung?

- (1) Positive Kantengewichte
- (2) Symmetrische Kantengewichte
- (3) Stark zusammenhängender Graph
- (4) Zusammenhängender Graph
- (5) Kreisfreier Graph
- (6) Keine dieser Bedingungen
- (7) Keine Ahnung

Auflösung: (6)

Paarweises Alignment (1) demogr.

In welcher Laufzeit wird das optimale paarweise Alignment für zwei Strings mit Längen n und m berechnet?

- (1) $\Theta(n + m)$
- (2) $\Theta((\min\{n, m\})^2)$
- (3) $\Theta(n \cdot m)$
- (4) $\Theta((\max\{n, m\})^2)$
- (5) $\Theta(n^m)$

Auflösung: (3) $\Theta(n \cdot m)$

Paarweises Alignment (2) demogr.

Wie schnell kann man das beste Alignment finden, in dem bei jedem String höchstens k Blanksymbole benutzt werden?

- (1) $\Theta(\min\{n, m\} \cdot k)$
- (2) $\Theta((n \cdot m)/k)$
- (3) $\Theta(n \cdot m \cdot \log k)$
- (4) $\Theta(n \cdot m \cdot k)$
- (5) $\Theta((n \cdot m)^k)$

Auflösung: (1) $\Theta(\min\{n, m\} \cdot k)$

APSP demogr.

Betrachte die Klasse von gerichteten Graphen G mit n Knoten, $m = \Theta(n^{1.5})$ Kanten und reellen Kantengewichten.

Welcher Algorithmus hat die schnellste worst-case Laufzeit für das APSP Problem in dieser Klasse von Graphen?

- (1) Bellman-Ford
- (2) Floyd
- (3) $n \times$ Dijkstra
- (4) (1) und (2) sind beide gleich schnell
- (5) Alle gleich gut

Auflösung: (2) Floyd

Paarweises Alignment demogr.

In welcher Laufzeit wird das optimale paarweise Alignment für zwei Strings mit Längen n und m berechnet?

- (1) $\Theta(n + m)$
- (2) $\Theta((\min\{n, m\})^2)$
- (3) $\Theta(n \cdot m)$
- (4) $\Theta((\max\{n, m\})^2)$
- (5) $\Theta(n^m)$

Auflösung: (3) $\Theta(n \cdot m)$

RNA Sekundärstruktur demogr.

Wie schnell kann man die optimale Sekundärstruktur mit maximaler Anzahl an Bindungen finden?

- (1) $\Theta(n)$
- (2) $\Theta(n^2)$
- (3) $\Theta(n^3)$
- (4) $\Theta(n^4)$
- (5) $\Theta(n^5)$

Auflösung: (3) $\Theta(n^3)$