Approximationsalgorithmen (1) multiple choice

Welche der folgenden Aussagen stimmen (mit unserer Definition von Approximationsfaktoren)?

- (1) Je kleiner der Approximationsfaktor δ, umso besser ist die im Worst-Case gefundene Lösung.
- (2) Je größer der Approximationsfaktor δ, umso besser ist die im Worst-Case gefundene Lösung.
- (3) $0 < \delta < 1$
- (4) $\delta \geq 1$

Approximationsalgorithmen (1) multiple choice

Welche der folgenden Aussagen stimmen (mit unserer Definition von Approximationsfaktoren)?

- (1) Je kleiner der Approximationsfaktor δ, umso besser ist die im Worst-Case gefundene Lösung.
- (2) Je größer der Approximationsfaktor δ, umso besser ist die im Worst-Case gefundene Lösung.
- (3) $0 < \delta < 1$
- (4) $\delta \geq 1$

Auflösung:

Approximationsalgorithmen (1) multiple choice

Welche der folgenden Aussagen stimmen (mit unserer Definition von Approximationsfaktoren)?

- (1) Je kleiner der Approximationsfaktor δ, umso besser ist die im Worst-Case gefundene Lösung.
- (2) Je größer der Approximationsfaktor δ, umso besser ist die im Worst-Case gefundene Lösung.
- (3) $0 < \delta < 1$
- (4) $\delta \geq 1$

Auflösung: (1) & (4)

Der Online Greedy-Scheduling Algorithmus aus der Vorlesung ist

- (1) $(1/2)$-approximativ
- (2) $(3/2)$-approximativ
- (3) $(2-1/m)$-approximativ für m Maschinen
- (4) 2-approximativ
- (5) 3-approximativ

Approximationsalgorithmen (2) multiple choice

Der Online Greedy-Scheduling Algorithmus aus der Vorlesung ist

- (1) $(1/2)$-approximativ
- (2) $(3/2)$-approximativ
- (3) $(2 - 1/m)$-approximativ für m Maschinen
- (4) 2-approximativ
- (5) 3-approximativ

Auflösung:

Approximationsalgorithmen (2) multiple choice

Der Online Greedy-Scheduling Algorithmus aus der Vorlesung ist

- (1) $(1/2)$-approximativ
- (2) $(3/2)$-approximativ
- (3) $(2 - 1/m)$-approximativ für m Maschinen
- (4) 2-approximativ
- (5) 3-approximativ

Auflösung: (3) & (4) & (5)

Approximationsalgorithmen (3) multiple choice

Der Offline Greedy-Scheduling Algorithmus (mit Vorsortierung) aus der Vorlesung ist

- (1) 2-approximativ
- (2) $(3/2)$-approximativ
- (3) $(10/9)$-approximativ
- (4) $(1+\epsilon)$-approximativ

Der Offline Greedy-Scheduling Algorithmus (mit Vorsortierung) aus der Vorlesung ist

- (1) 2-approximativ
- (2) $(3/2)$-approximativ
- (3) $(10/9)$-approximativ
- (4) $(1+\epsilon)$-approximativ

Auflösung:

Approximationsalgorithmen (3) multiple choice

Der Offline Greedy-Scheduling Algorithmus (mit Vorsortierung) aus der Vorlesung ist

- (1) 2-approximativ
- (2) $(3/2)$-approximativ
- (3) $(10/9)$-approximativ
- (4) $(1 + \epsilon)$-approximativ

Auflösung: (1) & (2)

Approximationsalgorithmen (4) demogr.

Welche Laufzeit hat der exakte Algorithmus für eine Instanz des Rucksackproblems mit n Objekten und Wertesumme W?

- (1) $\Theta(n \cdot \log W)$
- (2) $\Theta(n \cdot W)$
- (3) $\Theta(n \cdot W!)$
- (4) $\Theta(n \cdot 1/W)$
- (5) $\Theta(\text{keine Idee})$

Welche Laufzeit hat der exakte Algorithmus für eine Instanz des Rucksackproblems mit n Objekten und Wertesumme W?

- (1) $\Theta(n \cdot \log W)$
- (2) $\Theta(n \cdot W)$
- (3) $\Theta(n \cdot W!)$
- (4) $\Theta(n \cdot 1/W)$
- (5) $\Theta(\text{keine Idee})$

Auflösung:

Welche Laufzeit hat der exakte Algorithmus für eine Instanz des Rucksackproblems mit n Objekten und Wertesumme W?

- (1) $\Theta(n \cdot \log W)$
- (2) $\Theta(n \cdot W)$
- (3) $\Theta(n \cdot W!)$
- (4) $\Theta(n \cdot 1/W)$
- (5) $\Theta(\text{keine Idee})$

Auflösung: (2) $\Theta(n \cdot W)$

Welche Laufzeit hat der $(1+\varepsilon)$-approximative Algorithmus für eine Instanz des Rucksackproblems mit n Objekten und Wertesumme W?

- (1) $\Theta(n^3/\varepsilon \cdot \log W)$
- (2) $\Theta(n^3/\varepsilon \cdot W)$
- (3) $\Theta(n^3/\varepsilon)$
- (4) $\Theta(n^2 \cdot \log 1/\varepsilon)$
- (5) Θ(gar / keine Idee)

Welche Laufzeit hat der $(1+\varepsilon)$-approximative Algorithmus für eine Instanz des Rucksackproblems mit n Objekten und Wertesumme W?

- (1) $\Theta(n^3/\varepsilon \cdot \log W)$
- (2) $\Theta(n^3/\varepsilon \cdot W)$
- (3) $\Theta(n^3/\varepsilon)$
- (4) $\Theta(n^2 \cdot \log 1/\varepsilon)$
- (5) Θ(gar / keine Idee)

Auflösung:

Approximationsalgorithmen (5) demogr.

Welche Laufzeit hat der $(1+\varepsilon)$-approximative Algorithmus für eine Instanz des Rucksackproblems mit n Objekten und Wertesumme W?

- (1) $\Theta(n^3/\varepsilon \cdot \log W)$
- (2) $\Theta(n^3/\varepsilon \cdot W)$
- (3) $\Theta(n^3/\varepsilon)$
- (4) $\Theta(n^2 \cdot \log 1/\varepsilon)$
- (5) Θ(gar / keine Idee)

Auflösung: (3) $\Theta(n^3/\varepsilon)$