

Es seien $f(n) = \sqrt{n} \cdot 2^{\sin(n)}$ und $g(n) = 10 \cdot 2^{\log_4 n}$. Was gilt dann?

- (1) $f(n) = O(g(n))$
- (2) $f(n) = o(g(n))$
- (3) $f(n) = \Omega(g(n))$
- (4) $f(n) = \omega(g(n))$
- (5) $f(n) = \Theta(g(n))$

Es seien $f(n) = \sqrt{n} \cdot 2^{\sin(n)}$ und $g(n) = 10 \cdot 2^{\log_4 n}$. Was gilt dann?

- (1) $f(n) = O(g(n))$
- (2) $f(n) = o(g(n))$
- (3) $f(n) = \Omega(g(n))$
- (4) $f(n) = \omega(g(n))$
- (5) $f(n) = \Theta(g(n))$

Auflösung:

Es seien $f(n) = \sqrt{n} \cdot 2^{\sin(n)}$ und $g(n) = 10 \cdot 2^{\log_4 n}$. Was gilt dann?

- (1) $f(n) = O(g(n))$
- (2) $f(n) = o(g(n))$
- (3) $f(n) = \Omega(g(n))$
- (4) $f(n) = \omega(g(n))$
- (5) $f(n) = \Theta(g(n))$

Auflösung: (1) & (3) & (5)

Es seien $f(n) = (\log_2 n)^{\log_2 n}$ und $g(n) = n^4 \log_2 n$. Was gilt dann?

- (1) $f(n) = O(g(n))$
- (2) $f(n) = o(g(n))$
- (3) $f(n) = \Omega(g(n))$
- (4) $f(n) = \omega(g(n))$
- (5) $f(n) = \Theta(g(n))$

Es seien $f(n) = (\log_2 n)^{\log_2 n}$ und $g(n) = n^4 \log_2 n$. Was gilt dann?

- (1) $f(n) = O(g(n))$
- (2) $f(n) = o(g(n))$
- (3) $f(n) = \Omega(g(n))$
- (4) $f(n) = \omega(g(n))$
- (5) $f(n) = \Theta(g(n))$

Auflösung:

Es seien $f(n) = (\log_2 n)^{\log_2 n}$ und $g(n) = n^4 \log_2 n$. Was gilt dann?

- (1) $f(n) = O(g(n))$
- (2) $f(n) = o(g(n))$
- (3) $f(n) = \Omega(g(n))$
- (4) $f(n) = \omega(g(n))$
- (5) $f(n) = \Theta(g(n))$

Auflösung: (3) & (4)

Betrachte die Darstellung eines Heaps im Array H .

Wo finden Sie den **rechten Kindknoten** von $H[3]$?

- (1) $H[1]$
- (2) $H[4]$
- (3) $H[6]$
- (4) $H[7]$

Betrachte die Darstellung eines Heaps im Array H .

Wo finden Sie den **rechten Kindknoten** von $H[3]$?

- (1) $H[1]$
- (2) $H[4]$
- (3) $H[6]$
- (4) $H[7]$

Auflösung:

Betrachte die Darstellung eines Heaps im Array H .

Wo finden Sie den **rechten Kindknoten** von $H[3]$?

- (1) $H[1]$
- (2) $H[4]$
- (3) $H[6]$
- (4) $H[7]$

Auflösung: (4) $H[7]$

Betrachte die Darstellung eines Heaps im Array H .

Wo finden Sie den **rechten Kindknoten** von $H[3]$?

- (1) $H[1]$
- (2) $H[4]$
- (3) $H[6]$
- (4) $H[7]$

Auflösung: (4) $H[7]$

Wo finden Sie den **Elternknoten** von $H[8]$?

- (1) $H[1]$
- (2) $H[4]$
- (3) $H[5]$
- (4) $H[9]$

Betrachte die Darstellung eines Heaps im Array H .

Wo finden Sie den **rechten Kindknoten** von $H[3]$?

- (1) $H[1]$
- (2) $H[4]$
- (3) $H[6]$
- (4) $H[7]$

Auflösung: (4) $H[7]$

Wo finden Sie den **Elternknoten** von $H[8]$?

- (1) $H[1]$
- (2) $H[4]$
- (3) $H[5]$
- (4) $H[9]$

Auflösung:

Betrachte die Darstellung eines Heaps im Array H .

Wo finden Sie den **rechten Kindknoten** von $H[3]$?

- (1) $H[1]$
- (2) $H[4]$
- (3) $H[6]$
- (4) $H[7]$

Auflösung: (4) $H[7]$

Wo finden Sie den **Elternknoten** von $H[8]$?

- (1) $H[1]$
- (2) $H[4]$
- (3) $H[5]$
- (4) $H[9]$

Auflösung: (2) $H[4]$

Position	1	2	3	4	5	6	7	8
Priorität	25	14	22	10	17	11	18	6

Für welches Indexpaar $\{i, j\}$ ist die Heap-Ordnung verletzt?

Position	1	2	3	4	5	6	7	8
Priorität	25	14	22	10	17	11	18	6

Für welches Indexpaar $\{i, j\}$ ist die Heap-Ordnung verletzt?

Auflösung:

Position	1	2	3	4	5	6	7	8
Priorität	25	14	22	10	17	11	18	6

Für welches Indexpaar $\{i, j\}$ ist die Heap-Ordnung verletzt?

Auflösung: $\{2, 5\}$, denn $H[2] < H[5]$

Position	1	2	3	4	5	6	7	8
Priorität	25	14	22	10	17	11	18	6

Für welches Indexpaar $\{i, j\}$ ist die Heap-Ordnung verletzt?

Auflösung: $\{2, 5\}$, denn $H[2] < H[5]$

Wie gelingt in diesem Fall die Reparatur?

- (1) `repair_down(2)`
- (2) `repair_up(2)`
- (3) `repair_down(5)`
- (4) `repair_up(5)`

Position	1	2	3	4	5	6	7	8
Priorität	25	14	22	10	17	11	18	6

Für welches Indexpaar $\{i, j\}$ ist die Heap-Ordnung verletzt?

Auflösung: $\{2, 5\}$, denn $H[2] < H[5]$

Wie gelingt in diesem Fall die Reparatur?

- (1) `repair_down(2)`
- (2) `repair_up(2)`
- (3) `repair_down(5)`
- (4) `repair_up(5)`

Auflösung:

Position	1	2	3	4	5	6	7	8
Priorität	25	14	22	10	17	11	18	6

Für welches Indexpaar $\{i, j\}$ ist die Heap-Ordnung verletzt?

Auflösung: $\{2, 5\}$, denn $H[2] < H[5]$

Wie gelingt in diesem Fall die Reparatur?

- (1) `repair_down(2)`
- (2) `repair_up(2)`
- (3) `repair_down(5)`
- (4) `repair_up(5)`

Auflösung: (1) und (4)

Tiefe eines Heaps mit n Knoten?

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

Tiefe eines Heaps mit n Knoten?

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

Auflösung:

Tiefe eines Heaps mit n Knoten?

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

Auflösung: (3) $\Theta(\log n)$

Tiefe eines Heaps mit n Knoten?

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

Auflösung: (3) $\Theta(\log n)$ Warum?