

```
for (int j = 1; j <= n; j = j * 2) {  
    print(j);  
}
```

Die Laufzeit in Abhängigkeit von n ist

- (1) $\Theta(\log n)$
- (2) $\Theta(n)$
- (3) $\Theta(n \log n)$
- (4) $\Theta(n^2)$
- (5) Θ ja, ich kann mich nicht entscheiden ...

```
for (int j = 1; j <= n; j = j * 2) {  
    print(j);  
}
```

Die Laufzeit in Abhängigkeit von n ist

- (1) $\Theta(\log n)$
- (2) $\Theta(n)$
- (3) $\Theta(n \log n)$
- (4) $\Theta(n^2)$
- (5) Θ ja, ich kann mich nicht entscheiden ...

Auflösung:

```
for (int j = 1; j <= n; j = j * 2) {  
    print(j);  
}
```

Die Laufzeit in Abhängigkeit von n ist

- (1) $\Theta(\log n)$
- (2) $\Theta(n)$
- (3) $\Theta(n \log n)$
- (4) $\Theta(n^2)$
- (5) Θ ja, ich kann mich nicht entscheiden ...

Auflösung: (1) $\Theta(\log n)$.

```
for (int i = 0; i < n; i++) {  
    for (int j = 0; j < 2*i; j++) {  
        print(j + i);  
    }  
}
```

Die Laufzeit in Abhängigkeit von n ist

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

```
for (int i = 0; i < n; i++) {  
    for (int j = 0; j < 2*i; j++) {  
        print(j + i);  
    }  
}
```

Die Laufzeit in Abhängigkeit von n ist

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

Auflösung:

```
for (int i = 0; i < n; i++) {  
    for (int j = 0; j < 2*i; j++) {  
        print(j + i);  
    }  
}
```

Die Laufzeit in Abhängigkeit von n ist

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

Auflösung: (4) $\Theta(n^2)$.

Was stimmt?

- (1) Jeder Graph G mit n Knoten und m Kanten hat einen Knoten v , so dass Breitensuche(v) eine Laufzeit $\Theta(\max(n, m))$ benötigt.
- (2) Für jeden Startknoten v in jedem Graphen G sind die Bäume von Breiten- und Tiefensuche unterschiedlich.
- (3) Für jeden Startknoten v in jedem Graphen G sind die Bäume von Breiten- und Tiefensuche gleich.
- (4) Für jede Kante in ungerichteten Graphen, die in einer Breitensuche eine Baumkante ist, unterscheiden sich die Generationen der Knoten um genau 1.
- (5) Für jede Kante in ungerichteten Graphen unterscheiden sich die Generationen der Knoten um höchstens 1.

Was stimmt?

- (1) Jeder Graph G mit n Knoten und m Kanten hat einen Knoten v , so dass Breitensuche(v) eine Laufzeit $\Theta(\max(n, m))$ benötigt.
- (2) Für jeden Startknoten v in jedem Graphen G sind die Bäume von Breiten- und Tiefensuche unterschiedlich.
- (3) Für jeden Startknoten v in jedem Graphen G sind die Bäume von Breiten- und Tiefensuche gleich.
- (4) Für jede Kante in ungerichteten Graphen, die in einer Breitensuche eine Baumkante ist, unterscheiden sich die Generationen der Knoten um genau 1.
- (5) Für jede Kante in ungerichteten Graphen unterscheiden sich die Generationen der Knoten um höchstens 1.

Auflösung:

Was stimmt?

- (1) Jeder Graph G mit n Knoten und m Kanten hat einen Knoten v , so dass Breitensuche(v) eine Laufzeit $\Theta(\max(n, m))$ benötigt.
- (2) Für jeden Startknoten v in jedem Graphen G sind die Bäume von Breiten- und Tiefensuche unterschiedlich.
- (3) Für jeden Startknoten v in jedem Graphen G sind die Bäume von Breiten- und Tiefensuche gleich.
- (4) Für jede Kante in ungerichteten Graphen, die in einer Breitensuche eine Baumkante ist, unterscheiden sich die Generationen der Knoten um genau 1.
- (5) Für jede Kante in ungerichteten Graphen unterscheiden sich die Generationen der Knoten um höchstens 1.

Auflösung: (4), (5)

Der **Algorithmus von Dijkstra** benötigt zur korrekten Ausführung

- (1) Nicht-negative Kantengewichte.
- (2) Kreisfreiheit bei negativen Kantengewichten.
- (3) Einen ungerichteten Graphen.
- (4) Keine dieser Bedingungen.

Der **Algorithmus von Dijkstra** benötigt zur korrekten Ausführung

- (1) Nicht-negative Kantengewichte.
- (2) Kreisfreiheit bei negativen Kantengewichten.
- (3) Einen ungerichteten Graphen.
- (4) Keine dieser Bedingungen.

Auflösung:

Der **Algorithmus von Dijkstra** benötigt zur korrekten Ausführung

- (1) Nicht-negative Kantengewichte.
- (2) Kreisfreiheit bei negativen Kantengewichten.
- (3) Einen ungerichteten Graphen.
- (4) Keine dieser Bedingungen.

Auflösung: (1)