Laufzeitanalyse (1) demogr.

```
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) Θja, ich kann mich nicht entscheiden ...

Laufzeitanalyse (1) demogr.

```
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) Θja, ich kann mich nicht entscheiden …

Auflösung:

Laufzeitanalyse (1) demogr.

```
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) Θja, ich kann mich nicht entscheiden . . .

Auflösung: (1) $\Theta(\log n)$.

Laufzeitanalyse (2) demogr.

```
for (int i = 0; i < n; i++) {
    for (int j = 0; j < 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 ...

Laufzeitanalyse (2) demogr.

```
for (int i = 0; i < n; i++) {
    for (int j = 0; j < 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 < 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) $\Theta(n^2)$.

```
int j;
for (i = 1; i <= n; i++) {
   j = 0;
   while (j < n) {
      j = j + i;
   }
}
```

Die Laufzeit in Abhängigkeit von n ist

- (1) $\Theta(\log n)$
- (2) $\Theta(n)$
- (3) $\Theta(n \log n)$
- (4) $\Theta(n^2)$
- (5) jetzt reichts aber ...

Laufzeitanalyse (3) demogr.

```
int j;
for (i = 1; i <= n; i++) {
    j = 0;
    while (j < n) {
        j = j + i;
    }
}
```

Die Laufzeit in Abhängigkeit von n ist

- (1) $\Theta(\log n)$
- (2) $\Theta(n)$
- (3) $\Theta(n \log n)$
- (4) $\Theta(n^2)$
- (5) jetzt reichts aber …

Auflösung:

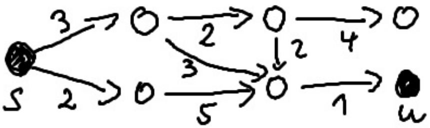
Laufzeitanalyse (3) demogr.

```
int j;
for (i = 1; i <= n; i++) {
    j = 0;
    while (j < n) {
        j = j + i;
    }
}
```

Die Laufzeit in Abhängigkeit von n ist

- (1) $\Theta(\log n)$
- (2) $\Theta(n)$
- (3) $\Theta(n \log n)$
- (4) $\Theta(n^2)$
- (5) jetzt reichts aber ...

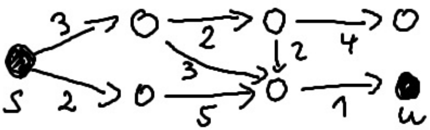
Auflösung: (3) $\Theta(n \log n)$.



Wie schnell kommt man von *s* nach *u*?

Der kürzeste Weg von *s* nach *u* hat Länge ...

Bsp. kürzeste Wege demogr.

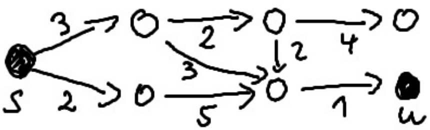


Wie schnell kommt man von s nach u?

Der kürzeste Weg von s nach u hat Länge ...

Auflösung:

Bsp. kürzeste Wege demogr.



Wie schnell kommt man von *s* nach *u*?

Der kürzeste Weg von *s* nach *u* hat Länge ...

Auflösung: 7.

Laufzeit Dijkstra-Algorithmus demogr.

Wenn wir einen binären Min-Heap als Datenstruktur benutzen, dann benötigt der Dijkstra-Algorithmus auf einem gerichteten Graphen mit n Knoten und m Kanten eine Laufzeit von

- (1) $\Theta(n + m)$
- (2) $\Theta(n \log n + m)$
- (3) $\Theta(n + m \log m)$
- (4) $\Theta((n + m) \cdot \log n)$
- (5) $\Theta(n \cdot m)$

Auflösung:

Laufzeit Dijkstra-Algorithmus demogr.

Wenn wir einen binären Min-Heap als Datenstruktur benutzen, dann benötigt der Dijkstra-Algorithmus auf einem gerichteten Graphen mit n Knoten und m Kanten eine Laufzeit von

- (1) $\Theta(n + m)$
- (2) $\Theta(n \log n + m)$
- (3) $\Theta(n + m \log m)$
- (4) $\Theta((n + m) \cdot \log n)$
- (5) $\Theta(n \cdot m)$

Auflösung: (4) $\Theta((n + m) \log n)$

Stabilität kürzester Wege demogr.

Was passiert mit den kürzesten Wegen für einen Graphen G, wenn bei jeder Kante des bisherigen Kürzeste-Wege Baumes T das Kantengewicht verdoppelt wird? Sei T' der neue Baum der kürzesten Wege.

- (1) T' kann sich von T unterscheiden.
- (2) T bleibt immer gleich.

Stabilität kürzester Wege demogr.

Was passiert mit den kürzesten Wegen für einen Graphen G, wenn bei jeder Kante des bisherigen Kürzeste-Wege Baumes T das Kantengewicht verdoppelt wird? Sei T' der neue Baum der kürzesten Wege.

- (1) T' kann sich von T unterscheiden.
- (2) T bleibt immer gleich.

Auflösung:

Stabilität kürzester Wege demogr.

Was passiert mit den kürzesten Wegen für einen Graphen G, wenn bei jeder Kante des bisherigen Kürzeste-Wege Baumes T das Kantengewicht verdoppelt wird? Sei T' der neue Baum der kürzesten Wege.

- (1) T' kann sich von T unterscheiden.
- (2) T bleibt immer gleich.

Auflösung: (1)