Manchmal hat man nur untere oder obere Abschätzungen:

DEFINITION 3.10 (O-Notation)

Seien $f, g : \mathbb{N} \to \mathbb{R}$ Funktionen.

Dann gilt $f \in \Theta(g) :\Longleftrightarrow$ Es gibt positive Konstanten c, n_0 mit

$$0 \leq f(n) \leq c\,g(n) \quad \text{für alle } n \geq n_0.$$

Man sagt: f wächst ~~exponentiell~~ höchstens in derselben Größenordnung wie g.

Beispiele:
$$2n^2-1 \in O(n^2)$$
$$2n^2-1 \in O(n^3)$$
$$n \log n \in O(n^2)$$

DEFINITION 3.11 (Ω-Notation)

Seien $f, g : \mathbb{N} \to \mathbb{R}$ Funktionen.

Dann gilt $f \in \Omega(g) :\Longleftrightarrow$ Es gibt positive Konstanten c, n_0 mit

$$0 \leq c\,g(n) \leq f(n) \quad \text{für alle } n \geq n_0.$$

Beispiele: $2n^2-1 \in \Omega(n^2)$
$2n^3-1 \in \Omega(n^2)$

Einige einfache Eigenschaften:

SATZ 3.12

Seien $f, g : \mathbb{N} \to \mathbb{R}$ Funktionen.

Dann gilt

(1) $f \in \Theta(g) \Longleftrightarrow g \in \Theta(f)$

(2) $f \in \Theta(g) \Longleftrightarrow f \in O(g)$ und $f \in \Omega(g)$.

(3) $f \in O(g) \Longleftrightarrow g \in \Omega(f)$

Beweis: Übung!

3.8 Laufzeit von BFS und DFS

Wenn man einen Graphen durchsucht, sollte man schon alle Koten und alle Kanten ansehen; also lässt sich eine untere Schranke von $\Omega(n+m)$ nicht unterbieten.
Tatsächlich wird diese Schranke auch erreicht:

SATZ 3.13

Der Graphen-Scan-Algorithmus 3.7 lässt sich so implementieren, dass die Laufzeit $O(n+m)$ ist.

Beweis:

Wir nehmen an, dass G durch eine Adjazenzliste gegeben ist.

Für jeden Knoten verwenden wir einen Zeiger, der auf die „aktuelle" Kante für diesen Knoten in der Liste zeigt (d.h. auf den „aktuellen" Nachbarn).
Anfangs zeigt akt(x) auf das erste Element in der Liste.
In 2.3.1 wird der aktuelle Nachbar ausgewählt und der Zeiger weiterbewegt; wird das Listenende erreicht, wird x aus R entfernt und nicht mehr eingeführt.
Also ergibt sich eine Gesamtlaufzeit von $O(n+m)$. $\square$

KOROLLAR 3.14

Mit Algorithmus 3.7 kann man die Zusammenhangskomponenten eines Graphen berechnen.

Beweis:

Wende 3.7 an und überprüfe, ob $Y = V$ ist. Falls ja, ist der Graph zshgd.
Falls nein, haben wir eine Zusammenhangskomponente berechnet;
wir lassen den Algorithmus erneut für einen Knoten $s' \in V \setminus Y$ laufen, usw.
Wieder wird keine Kante von einem Knoten aus doppelt angefasst,
also bleibt die Laufzeit linear, d.h. $O(n+m)$. $\square$