

Die Potenzfunktion

C/C++ besitzen schon eine Potenzfunktion `pow(x, e)`, die x^e berechnet. Allerdings sind nicht alle Datentypen möglich (dafür aber auch reelle Exponenten) und die Funktion ist derzeit nicht `constexpr`, d.h. ihre Werte können nicht schon vom Compiler berechnet werden. Ich zeige hier Möglichkeiten, wie man die Potenz für beliebige Basis-Typen und positive ganzzahlige Exponenten realisieren kann.

Mittels einer Schleife und Multiplikationen:

```
template<typename T>
constexpr T power(const T& x, int e)    // funktioniert nur bei e >= 0
{
    T prod{1};
    for (int i = 0; i < e; ++i)
        prod = x*prod;    // falls T keinen operator*=( ) besitzt
    return prod;
}
```

Der Rechenaufwand steigt linear mit dem Exponenten e : $O(e)$

Mittels Rekursion:

Dank der 2 Beziehungen $x^{e+1} = x * x^e$ und vor allem $x^{2^e} = (x^2)^e$ kann man die Potenzfunktion kurz und effizient per Rekursion definieren:

```
template<typename T>
constexpr T power(const T& x, int e)    // funktioniert nur bei e >= 0
{
    if (e <= 1)
        return e ? x : T{1};
    return e % 2 == 0 ? power(x*x, e/2) : x * power(x, e-1);
}
```

Diese Funktion ist nicht nur kürzer, sondern auch viel schneller: Der Rechenaufwand steigt logarithmisch: $O(\log(e))$

Mittels einer schnellen Schleife und Multiplikationen:

Man kann die Ideen der Rekursionsmethode auch als Schleife realisieren und so eine nichtrekursive Funktion mit Aufwand $O(\log(e))$ erzeugen.