# **Нотация Big-O**

Използваме нотация big-Θ за да ограничим асимптотично отгоре и отдолу растежа на времето за изпълнение в рамките на константните коефициенти. Понякога искаме да ограничаваме само отгоре.

Например въпреки че времето за изпълнение на двоично търсене в най-лошия случай е *\Theta(\log\_2 n)*Θ(log2 *n*)\Theta, left parenthesis, log, start base, 2, end base, n, right parenthesis, би било неправилно, ако заявим, че двоичното търсене се изпълнява за време *\Theta(\log\_2 n)*Θ(log2 *n*)\Theta, left parenthesis, log, start base, 2, end base, n, right parenthesis във *всички* случаи. Ами ако открием търсената стойност с първото си предположение? Тогава алгоритъмът се изпълнява за време *\Theta(1)*Θ(1)\Theta, left parenthesis, 1, right parenthesis. Времето за изпълнение на двоичното търсене никога не е по-бавно от *\Theta(\log\_2 n)*Θ(log2 *n*)\Theta, left parenthesis, log, start base, 2, end base, n, right parenthesis, но понякога е по-бързо.

Би било удобно, ако имаме вид асимптотична нотация, която означава, че "времето за изпълнение нараства най-много толкова, но може бавно да порасне още". Използваме "big-O" нотация за точно такива случаи.

Ако времето за изпълнение е *O(f(n))O*(*f*(*n*))O, left parenthesis, f, left parenthesis, n, right parenthesis, right parenthesis, то за достатъчно голямо *nn*n времето за изпълнение е най-много *k \cdot f(n)k*⋅*f*(*n*)k, dot, f, left parenthesis, n, right parenthesis за константа *kk*k. Ето как да мислим за време *O(f(n))O*(*f*(*n*))O, left parenthesis, f, left parenthesis, n, right parenthesis, right parenthesis: