**existing library function pow in python uses binary exponentiation?**

Yes, Python's built-in **pow** function uses an efficient algorithm—**binary exponentiation**—especially for modular exponentiation with three arguments (pow(base, exp, mod)), and in general for integer powers, providing significant speed advantages over naive repeated multiplication.[[1]](#fn1)[[2]](#fn2)[[3]](#fn3)

**How pow() Uses Binary Exponentiation**

* For large integer exponents and especially when the modulus is provided (three-argument form), pow() applies the binary exponentiation technique, drastically reducing time complexity from O(n) (naive) to O(log n).[[3]](#fn3)[[1]](#fn1)
* Binary exponentiation works by breaking down the exponent into binary form, squaring the base at each bit, and multiplying when the bit is set, making this ideal for large number and cryptographic computations.[[4]](#fn4)[[2]](#fn2)

**pow() Usage Highlights**

* pow(base, exp): Computes base raised to the power exp efficiently using binary exponentiation for integers.[[1]](#fn1)[[3]](#fn3)
* pow(base, exp, mod): When a modulus is supplied, pow computes (base\*\*exp % mod) using repeated squaring (binary exponentiation) and modulus at each step, which is highly efficient for both speed and memory—especially relevant for cryptography and competitive programming.[[3]](#fn3)[[1]](#fn1)
* For floating-point numbers, pow uses underlying math libraries and does not apply binary exponentiation.[[2]](#fn2)

**Efficiency Comparison**

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Time Complexity | Used By pow() | Efficiency Benefits |
| Naive Multiplication | O(n) | No | Slow for big exponents [[5]](#fn5) |
| Binary Exponentiation | O(log n) | Yes (integers/modular) | Fast and memory-efficient [[4]](#fn4)[[2]](#fn2) |

Python's pow function is optimized for **fast exponentiation using binary exponentiation**, especially for large integer and modular calculations.[[2]](#fn2)[[1]](#fn1)[[3]](#fn3)

⁂

1. <https://docs.vultr.com/python/built-in/pow>

1. <https://www.geeksforgeeks.org/python/fast-exponentiation-in-python/>

1. <https://realpython.com/ref/builtin-functions/pow/>

1. <https://cp-algorithms.com/algebra/binary-exp.html>

1. <https://www.freecodecamp.org/news/binary-exponentiation-algorithm-explained-with-examples/>

1. <https://www.geeksforgeeks.org/python/python-pow-function/>

1. <https://www.geeksforgeeks.org/competitive-programming/binary-exponentiation-for-competitive-programming/>

1. <https://www.keyskillset.com/resources/how-to-use-the-pow-function-in-python-examples-and-syntax>

1. <https://unstop.com/blog/python-pow-function>

1. <https://takeuforward.org/data-structure/implement-powxn-x-raised-to-the-power-n/>