- As pointed out by your father, for the master degree here, you only have one more year. To be or not to be, it's pretty obvious.
- The best worst-case running time for comparison sorting is $O(n \log n)$.

Master Theorem

The Master Theorem applies to recurrences of the following form:

$$T(n) = aT(n/b) + f(n)$$

where $a \geq 1$ and $b > 1$ are constants and $f(n)$ is an asymptotically positive function.

There are 3 cases:

1. If $f(n) = O(n^{\log_b a - \epsilon})$ for some constant $\epsilon > 0$, then $T(n) = \Theta(n^{\log_b a})$.
2. If $f(n) = \Theta(n^{\log_b a} \log^k n)$ with[1] $k \geq 0$, then $T(n) = \Theta(n^{\log_b a} \log^{k+1} n)$.
3. If $f(n) = \Omega(n^{\log_b a + \epsilon})$ with $\epsilon > 0$, and $f(n)$ satisfies the regularity condition, then $T(n) = \Theta(f(n))$.
   Regularity condition: $af(n/b) \leq cf(n)$ for some constant $c < 1$ and all sufficiently large n.

- Given an open-address hash table with load factor $\alpha = n/m < 1$, the expected number of probes in an unsuccessfully search is **at most** $1/(1-\alpha)$, assuming uniform hashing.

Table 3-1. *Some Basic Recurrences with Solutions, as Well as Some Sample Applications*

#	Recurrence	Solution	Example Applications
1	$T(n) = T(n-1) + 1$	$\Theta(n)$	Processing a sequence, for example, with `reduce`
2	$T(n) = T(n-1) + n$	$\Theta(n^2)$	Handshake problem
3	$T(n) = 2T(n-1) + 1$	$\Theta(2^n)$	Towers of Hanoi
4	$T(n) = 2T(n-1) + n$	$\Theta(2^n)$	
5	$T(n) = T(n/2) + 1$	$\Theta(\lg n)$	Binary search (see the black box sidebar on `bisect` in Chapter 6)
6	$T(n) = T(n/2) + n$	$\Theta(n)$	Randomized Select, average case (see Chapter 6)
7	$T(n) = 2T(n/2) + 1$	$\Theta(n)$	Tree traversal (see Chapter 5)
8	$T(n) = 2T(n/2) + n$	$\Theta(n \lg n)$	Sorting by *divide and conquer* (see Chapter 6)