

[Explore](#)[Problems](#)[Interview](#)

New

[Contest](#)[Discuss](#)[Store](#)[Premium](#)[Description](#)[Solution](#)[Discuss \(999+\)](#)[Submissions](#)

10. Regular Expression Matching

Hard[8462](#)[1292](#)[Add to List](#)[Share](#)

Given an input string `s` and a pattern `p`, implement regular expression matching with support for `'.'` and `'*'` where:

- `'.'` Matches any single character.
- `'*'` Matches zero or more of the preceding element.

The matching should cover the **entire** input string (not partial).

Example 1:

Input: `s = "aa", p = "a"`

Output: `false`

Explanation: "a" does not match the entire string "aa".

Example 2:

Input: `s = "aa", p = "a*"`

Output: `true`

Explanation: `'*'` means zero or more of the preceding element, `'a'`. Therefore, by repeating `'a'` once, it becomes "aa".

Example 3:

Input: `s = "ab", p = ".*"`

Output: `true`

Explanation: `".*"` means "zero or more (`*`) of any character (`.`)".

Constraints:

- `1 <= s.length <= 20`
- `1 <= p.length <= 30`
- `s` contains only lowercase English letters.
- `p` contains only lowercase English letters, `'.'`, and `'*'`.
- It is guaranteed for each appearance of the character `'*'`, there will be a previous valid character to match

[Problems](#)[Pick One](#)[Prev](#)

10/2323