

بخش محاسبه عبارت:

شامل دو تابع `in_to_post` و `evaluate` میباشد

### تابع `in_to_post`:

شامل توابع `stack::is-empty`, `stack::push`, `precedence` که مرتبه زمانی هر سه  $O(1)$  میباشد و تابع `qstring::contains` که دارای مرتبه زمانی  $O(1)$  میباشد (در حالت کلی مرتبه زمانی  $n$  دارد اما در این حالت نهایت تکرار حلقه `for` داخل تابع مشخص و برابر 8 میباشد در نتیجه تأثیری در مرتبه زمانی کل تابع ندارد)

دارای حلقه `for` که به اندازه سائز رشته تکرار میشود و مرتبه زمانی آن  $O(n)$  میباشد

پس مرتبه زمانی کل تابع  $O(n)$  میباشد.

### تابع `evaluate`:

شامل توابع `append()`, `stack::push`, `stack::pop` که دارای مرتبه زمانی  $O(1)$  هستند.

شامل تابع `post_to_in` که دارای حلقه `for` به اندازه سائز رشته است و مرتبه زمانی  $O(n)$  دارد.

شامل تابع `qstring::contains` که دارای مرتبه زمانی  $O(1)$  میباشد (در حالت کلی مرتبه زمانی  $n$  دارد اما در این حالت نهایت تکرار حلقه `for` داخل تابع مشخص و برابر 9 میباشد در نتیجه تأثیری در مرتبه زمانی کل تابع ندارد).

شامل توابع `indexOf()`, `replace()`, `qstring::number` که هر سه مرتبه زمانی  $O(n)$  دارند.

شامل حلقه که به اندازه لیست ورودی تکرار خواهد شد و شامل 3 تابع بالا هست پس مرتبه زمانی برابر  $O(n^2)$  است.