Algorithm 1: bisect right in rotated sorted array

```
Data: start searching index $lo$, end searching index $hi$, element to
      locate $x$, rotated sorted array $X_1, ..., X_N$
Result: position of element to locate $p$
Initialize $lo_{backup} \leftarrow lo$;
Initialize $hi_{backup} \leftarrow hi$;
Initialize $p \leftarrow 0$;
if $X_{hi-1} \leq x < X_{lo}$ then
    $p \leftarrow hi - 1$;
else
    while $lo < hi$ do
        $mid \leftarrow (lo + hi)/2$;
        if $X_{lo} < X_{mid}$ then
            if $X_{mid} \leq x$ then
                $lo \leftarrow mid$;
            else
                if $X_{lo} \leq x < X_{mid}$ then
                    $hi \leftarrow mid$;
                else
                    $lo \leftarrow mid$;
                end
            end
        else if $X_{lo} == X_{mid}$ then
            if $hi - lo == 1$ then
                $hi \leftarrow mid$;
            else
                $lo \leftarrow mid$;
            end
        else
            if $x < X_{mid}$ then
                $hi \leftarrow mid$;
            else
                if $X_{mid} \leq x < X_{lo}$ then
                    $lo \leftarrow mid$;
                else
                    $hi \leftarrow mid$;
                end
            end
        end
    end
    $p \leftarrow hi$;
end
return $max(min(hi_{backup}, p), lo_{backup})$
```