Exercise: Pow

2/4/2024

1/1 Points

Attempt 2

Review Feedback
1/24/2024

Attempt 2 Score:
1/1

View Feedback

Anonymous Grading: **no**

Unlimited Attempts Allowed
1/19/2024 to 2/4/2024

Details

Consider the following algorithm for determining x^n for non-negative n.

```
func pow(x, n):
  result = 1
  do n times:
    result *= x
  return result
```

A. Provide pseudocode for a recursive implementation of the above algorithm.
B. How many multiplications does this algorithm perform? Justify your answer.
C. Provide pseudocode for a recursive algorithm that computes x^n with $\mathcal{O}(\log n)$ multiplications.
D. Provide pseudocode for an iterative version of the $\mathcal{O}(\log n)$ algorithm.

View Rubric

Pow

Criteria	Ratings		Pts
(a) correct algorithm	**0.25 pts** **Full Marks**	**0 pts** **No Marks**	/ 0.25 pts
(b) correct answer and justification	**0.25 pts** **Full Marks**	**0 pts** **No Marks**	/ 0.25 pts
(c) correct algorithm (must be recursive)	**0.25 pts** **Full Marks**	**0 pts** **No Marks**	/ 0.25 pts
(d) correct algorithm (must be iterative)	**0.25 pts** **Full Marks**	**0 pts** **No Marks**	/ 0.25 pts

Total Points: 0

A.

```
function rec_pow(x, n):
  if n == 0:
    return 1
  else:
    return x * rec_pow(x, n - 1)
```

B.

There are n multiplications.

Proceed by mathematical induction on n.

When n = 0, zero multiplications are done.

Assume when n = k, k multiplications are done.

Then, when n = k + 1.

pow(x, k+1) = x*pow(x, k)

This has one multiplication as well as the multiplications in pow(x, k).

By Inductive Hypothesis, pow(x, k) performs k multiplications. Altogether, this is k+1 multiplications total.

C.

```
function recursiveExp(x, n):
    if n is 1:
        return x
    power = recursiveExp(x, floor(n / 2))
    If n is even:
        return power * power
    else if n is odd:
        return power * power * x
```

D.

```
function iterativeExp(x, n):
    result = 1
    while n > 0:
        if n is odd:
            result *= x
        x *= x
        n = floor(n / 2)
    return result
```