

Q1 Give an algorithm (pseudo code, with explanation) to compute 2^{2^n} in linear time, assuming multiplication of arbitrary size integers takes unit time. What is the bit-complexity if multiplications do not take unit time, but are a function of the bit-length.

Solution:

Algorithm to compute 2^{2^n} in linear time.

`mult(n):`

 Inputs: the value n

 Outputs: the product

 x = 2 # base case

 for i in n:

 x = x squared

 return x

Assuming that arbitrary size integers multiplication take unit time, we have $arr[i-1]$ multiplied by $arr[i-1]$. If they are $O(1)$ and the loop runs n times, then we have $O(n)$, which is linear time.