Imperial College London – Department of Computing

MSc in Computing Science

580: Algorithms
Tutorial 2

1. (Part of a 2015 exam question.)

   (a) Using either Java or pseudocode, write a recursive procedure Pow(x, N) to compute x^N, where N is a positive integer. Use a divide and conquer strategy. *Hint:*

   $$x^N = x^{N/2} \times x^{N/2} \qquad \text{for even } N$$
   $$x^N = x^{(N-1)/2} \times x^{(N-1)/2} \times x \qquad \text{for odd } N.$$

   (b) Write recurrence expressions for the time complexity $T(N)$ of your Pow procedure in the following cases:

   What is

   (c) Solve your expressions for $T(N)$ using the ma

Answer:

(a)

```
1:  procedure Pow(x, n)
2:      if n == 1 then
3:          return x
4:      end if
5:      tmp = Pow(x, n/2)                          ▷ integer division
6:      if n mod 2 == 0 then
7:          return tmp × tmp
8:      end if
9:      return tmp × tmp × x
10: end procedure
```

Key feature: Pow should be called recursively ONCE, in line 5. If it is called more than once then the complexity will be $\Theta(N)$.

(b)

$$T(N) = \begin{cases} \Theta(1) & \text{, if } N = 1 \\ T(\lfloor N/2 \rfloor) + \Theta(1) & \text{, if } N > 1 \end{cases}$$

So, $c = 1$. Symbols representing constants would be acceptable in place of the $\Theta(1)$ terms. The second formula must include $\lfloor N/2 \rfloor$, not $N/2$. The expression is not correct for all N if it uses $T(N/2)$.

(c) To apply the master method you need to first substitute the recurrence into the expression $T(N) = aT(N/b) + f(N)$, while ignoring any floors or ceilings. This gives us $a = 1$, $b = 2$ and $f(N) = \Theta(1)$. Then compute $N^{\log_b a}$, which in this case is $N^0 = 1$. So, $N^{\log_b a} = \Theta(1)$ and $f(N) = \Theta(1) = \Theta(N^{\log_b a} \log_2^0 N)$. Therefore, Case 2 of the master method applies and $T(N) = \Theta(N^{\log_b a} \log_2 N) = \Theta(\log_2 N)$.