Algorithm Analysis and Design

Ghazale Kharadpoor 9513012

October, 2018

- The *expression* must be the variable a.

This algorithm computes the exponentiation of a variable such x in an optimized way. In a case like $Pow(x, k)$, the algorithm prefer to calculate the square of the current number instead of multiplying it k times. It analyse the given number k as a binary number and traverse the binary number from the right side to the left. If the unseen digit from the right side of binary form of the k is 1 then the algorithm consider it as an odd number. In this case, we multiply a to y (which squares in every loop). Actually we multiply a to y^i which i is the steps of traversed digits. If the digit is 0, then we just square y and then go to the next digit.

Consider $x = 3$ and $k = 23$. In this case, the binary form of 23 is 10111.
So the result must be $3^{2^0} \times 3^{2^1} \times 3^{2^2} \times 3^{2^4}$.

- The loop of the algorithm is just the one part that gets some times. The loop traverse the binary number digits and the length of this number is $\log n$. So time complexity of this algorithm is $O(\log n)$.

- At the j-th iteration of the while loop, we are traversing the j-th digit from the right side of the binary number from the given x in $Pow(x, k)$. At this step, a is equal to x^i which i is decimal form of j-1 traversed digits of binary number.

Initialization: In this step we have to show the loop invariant is true prior to the first iteration of the while loop. Before any iteration of the while loop (that means before any traverses of the binary number of k) $a = 1$ that is right because $1 = x^i$ that $i = 0$; So that $a = x^0$ ✓
Also for y, because we didn't pass any loops, so we didn't traverse any digits of the binary number and $y = x^{2^0}$. ✓

Maintenance: In this step, we have to show the loop invariant is true before and after one iteration of the while loop. Before the j-th iteration of the while loop, $a = x^i$ which i is represented of the decimal value of j-1 traversed

binary digits and also $y = x^{2^{j-1}}$.
After the iteration, if the j-th digit of the binary number is 1 ($a \leftarrow a \times y$), then $a = x^{i+2^{j-1}}$ which It has to represent the decimal form of traversed binary number. i was the decimal of previous traversed binary number and now 2^{j-1} is represented the next decimal digit of the binary number.
For y ($y \leftarrow y \times y$), We know that after one iteration of the loop, y must be squared. As we see, $y = x^{2^j}$. ✓

Termination: When the loop terminates, $i \leq 0$. Because in all steps i is divided to 2 and the whole number of iterations is equal to $logk$ which this is the number of binary form of k digits. At this case, we traversed all of the binary digits of k and a is equal to x^k and this is what we wanted from the algorithm. So the algorithm is correct and gives what we wanted from it.