Introduction To Algorithms
HomeWork 1 Solutions

Junzhe Zheng

January 25, 2014

1. I used Matlab to solve this problem and got result is 30.
Code:

```
n=1;
while 200*n^2 > 1.5^n
    n = n + 1;
end
disp(n);
```

2. (a) is False.
For example, $f(n) = log(n)$ and $g(n) = n^3$. By assuming that $g(n) = \hat{O}(f(n))$, we have:

$$0 \leq n^3 \leq c(log(n))^2, \quad \forall n > n_0$$

For all $n > 1$, this does not hold. Thus the assumption is wrong,

(b) is conditional True.
$f(n) = \hat{O}(g(n))$, by definition, we have:

$$0 \leq f(n) \leq cg(n)log(n), \quad \forall n > n_0$$

For $f(n) \geq 1$,taking the logarithm on both side, we get:

$$0 \leq log(f(n)) \leq logc + log(g(n)) + log(log(n)), \quad \forall n > n_0$$

If $log(g(n)) \geq 1$ for sufficiently large n, we will have:

$$0 \leq log(f(n)) \leq logc * log(g(n)) + log(g(n)) + log(g(n)) * log(log(n)), \quad \forall n > n_0$$

Because n is very large, then $log(n) \geq log(log(n))$ and $log(n) \geq 1$, we get

$$0 \leq log(f(n)) \leq logc*log(g(n))*log(n)+log(g(n))*log(n)+log(g(n))*log(n), \quad \forall n > n_0$$

then:

$$0 \leq log(f(n)) \leq (logc + 2) * log(g(n)) * log(n), \quad \forall n > n_0$$
$$0 \leq log(f(n)) \leq c_2 * log(g(n)) * log(n), \quad \forall n > n_0$$

By definition, this implies $log(f(n)) = \hat{O}(log(g(n)))$.
Thus, this is True for $f(n) \geq 1$,$log(n) \geq 1$ and n is a very large number.

(c) is True.
By definition, $o(f(n))$ implies:

$$0 \leq g(n) < cf(n), \quad \forall n > n_0 \quad and \quad \forall c > 0$$

Both sides plus $f(n) > 0$, we get:

$$0 \leq f(n) \leq f(n) + g(n) < (c+1)f(n), \quad \forall n > n_0 \quad and \quad \forall c > 0$$

$$0 \leq f(n) \leq f(n) + g(n) < c_2 f(n), \quad \forall n > n_0 \quad and \quad \forall c_2 > 1$$

But we can choose a smaller c_2 to make:

$$0 \leq f(n) \leq f(n) + g(n) \leq c_2 f(n), \quad \forall n > n_0 \quad and \quad \forall c_2 > 0$$

By definition, this implies $f(n) + g(n) = \Theta(f(n))$.

(d) is False.
Let $f(n) = 1$ and $g(n) = n$.Then $f(n) + g(n) = 1 + n \leq O(min(f(n), g(n))) = O(f(n)) = cf(n) = c$,for $\forall n \geq n_0$,which is impossible.

3. For a N floor building, drop first egg at floor x. This can lead to two results:1.Break. Then we should start at ground floor to x floor, worst case is x. 2 Not Break. Then we should go up x-1 floor from x, which leads us to two situations. If break, then start at x+1 floor to x+(x-1) floor, worst case x-1. But in our previous test, we had dropped egg at x floor , so the worst case for this is still x-1+1=x. If Not break, we just recursively do above procedure with 1 less floor when go up till x =1. Here we have:

$$n + (n-1) + (n-2) + \cdots + 3 + 2 + 1 \geq N$$

we solve for:

$$\frac{(n+1)*n}{2} = N$$

Because floor number is a positive integer, solution is:

$$n = \lceil \frac{-1 + \sqrt{1+8N}}{2} \rceil$$