Illinois Institute of Technology
Department of Computer Science

Homework 1 Solutions

CS 430 Introduction to Algorithms
Spring Semester, 2014

1. **Problem 1 Solution:** The smallest value of n such that an algorithm whose running time is $200n^2$ runs faster than an algorithm whose running time is 1.5^n is 30.

2. **Problem 2. Solution:**

   (a) False. Let $f(n) = n$ and $g(n) = n^2$, then for some constant c we have
   $$0 \le n \le cn^2 \log n$$
   $$f(n) = \hat{O}(g(n))$$
   Suppose that $g(n) = \hat{O}(f(n))$, then for some constant c' we have
   $$0 \le g(n) \le c' f(n) \log n$$
   $$0 \le n^2 \le c' n \log n$$
   However, there exists n_0 such that $n^2 > c' n \log n$ when $n > n_0$. Thus $f(n) = \hat{O}(g(n))$ does not imply $g(n) = \hat{O}(f(n))$

   (b) True. Since $f(n) = \hat{O}(g(n))$,
   we have
   $$0 \le f(n) \le cg(n)\ \log n \text{ and}$$
   $$0 \le \log f(n) \le \log(cg(n) \log n) = \log c + \log g(n) + \log \log n$$
   Because
   $$\log c + \log g(n) + \log \log n \le c \log g(n) \log n,$$
   for some constant c when $n > n_0$, we have
   $$0 \le \log f(n) \le c \log g(n) \log n,$$
   which implies $\log f(n) = \hat{O}(\log g(n))$

   (c) Let $h(n) = o(f(n))$, then $\exists c > 0$, such that $h(n) < cf(n)$ and
   $$f(n) \le f(n) + h(n) < (1 + c)f(n).$$
   which implies
   $$f(n) + o(f(n)) = \theta(f(n))$$

   (d) False. Let $f(n) = n$ and $g(n) = n^2$, then $min(f(n), g(n)) = n$. However $f(n) + g(n) = n^2 \neq n$.

3. **Problem 3. Solution:** With the first egg, we can go up say n floors at a step. If the first egg breaks, we can only go up one floor at a time from the previous step with the second egg. To minimize the number of tests, we need to balance the big steps and the small steps. We can formulate this process as a decision tree and make the tree balanced.

   As we can see from the above decision tree, if we go for big steps of sizes $n, 2n1, 3n2, ,$ then we balance the height of the tree to n. Let h be the height of the building, then we have
   $n + (n - 1) + (n - 2) + ... + 1 = \frac{n(n+1)}{2} \ge h$
   By solving the above inequation we can get the optimal step sizes.

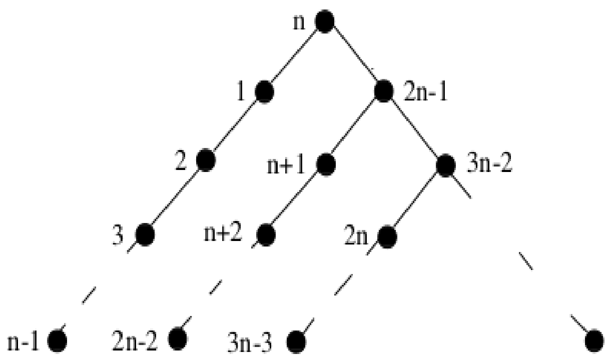

Figure 1: The graph