Exercise Questions Lecture 1-8: PS-2

Suchintan Mishra

Department of Computer Science and Engineering
Institute of Technical Education and Research,
Siksha 'O' Anusandhan (Deemed to be University), Bhubaneswar, Odisha, India

Text Books

Text book and References:

- (T1) Algorithm Design by Jon Kleinberg and Eva Tardos, Pearson Publication Reference book
- (R1) The Algorithm Design Manual by Steven Skiena, Springer Publication
- (R2) Introduction to Algorithms by CLRS, PHI Publication

Outline

Give Counter Examples: Steven Skienna Page 27

1-1. *[3]* Show that $a + b$ can be less than $\min(a, b)$.

1-2. *[3]* Show that $a \times b$ can be less than $\min(a, b)$.

Steven Skienna Page 28

1-7. *[3]* Prove the correctness of the following recursive algorithm to multiply two natural numbers, for all integer constants $c \geq 2$.

function multiply(y, z)

comment Return the product yz.

1. *if* $z = 0$ *then* return(0) *else*
2. return(multiply$(cy, \lfloor z/c \rfloor) + y \cdot (z \bmod c)$)

Steven Skienna Page 28

1-10. *[3]* Prove that $\sum_{i=1}^{n} i = n(n+1)/2$ for $n \geq 0$, by induction.

1-11. *[3]* Prove that $\sum_{i=1}^{n} i^2 = n(n+1)(2n+1)/6$ for $n \geq 0$, by induction.

1-13. *[3]* Prove that

$$\sum_{i=1}^{n} i(i+1)(i+2) = n(n+1)(n+2)(n+3)/4$$

Cormen Page 11 I

Other than speed, what other measures of efficiency might one use in a real-world setting?

Cormen Page 11 II

There are various criteria used as a measurement of efficient software; these include,

- **Space usage** that contains memory utilization.
  Software that uses less memory would be compatible for economic devices with lower memory, and would, therefore, yield better sales. It would be, even, reflected on the speed of execution because the software might be suspended until there is available memory for it. Space usage includes other storage requirements like hard drive requirement.
- Bandwidth requirement and utilization that is an element of **network utilization**.
  Just like space usage, the less the software uses a network the better it is because it would not wait until the network is available.
- The number of database transactions per second and their type, or generally **database requirement**.
  Again, the software would not wait for the database until it is available to serve its request.
- Special **hardware requirement**, like whether using specific pieces of computation or a general one.
  special hardware may yield a great enhancement to the speed, but it would increase the price.
- Programming languages.
  In real-world challenges, you may prefer one programming language to another one for various reasons like the available support.
- Simplicity and readability.
  You may think something like simplicity has nothing to efficiency because it would not enhance the speed—it may even decrease the speed of execution. Simpler solutions are popular and efficient because anyone can understand it and, therefore, add more features in less development time, which is a lot better for business goals than complex solutions that may be ready after years of development. For some applications, Java is more popular than C++ regardless that C++ has a great speed advantage over Java.
- Maintainability and better design.
  Tightly related to the previous point, software that is not maintainable is very bad for business. A good design would enable better maintainability and less development time to add more features.

Cormen Page 29

Express the function $n^3/1000 + 100n^2 - 100n + C$ in terms of θ-notation

Cormen Page 29

Consider sorting n numbers stored in array A by first finding the smallest element of A and exchanging it with the element in A[1]. Then find the second smallest element of A, and exchange it with A[2]. Continue in this manner for the first n-1 elements of A. Write pseudocode for this algorithm, which is known as selection sort. What loop invariant does this algorithm maintain? Why does it need to run for only the first n 1 elements, rather than for all n elements? Give the best-case and worst-case running times of selection sort in θ notation.

Algorithm 4 Selection Sort

```
for i = 1 to n - 1 do
    min = i
    for j = i + 1 to n do
        // Find the index of the i^th smallest element
        if A[j] < A[min] then
            min = j
        end if
    end for
    Swap A[min] and A[i]
end for
```

This yields a running time of

$$\sum_{i=1}^{n-1} n - i = n(n-1) - \sum_{i=1}^{n-1} i = n^2 - n - \frac{n^2 - n}{2} = \frac{n^2 - n}{2} = \Theta(n^2).$$

Cormen Page 53

Explain why the statement, "The running time of algorithm A is at least $O(n^2)$ is meaningless."

Let T(n) be the running time for algorithm A and let a function f(n) $= O(n^2)$. The statement says that T(n) is at least $O(n^2)$. That is, T(n) is an upper bound of f(n).

Cormen Page 53

Prove that $o(g(n)) \cap \omega(g(n))$ is the empty set.

USING CLASSICAL DEFINITIONS

By definition, $o(g(n))$ is the set of functions $f(n)$ such that $0 \leq f(n) < c_1 g(n)$ for any positive constant $c_1 > 0$ and all $n \geq n_0$.

And, $\omega(g(n))$ is the set of functions $f(n)$ such that $0 \leq c_2 g(n) < f(n)$ for any positive constant $c_2 > 0$ and all $n \geq n_0$.

So, $o(g(n)) \cap \omega(g(n))$ is the set of functions $f(n)$ such that,

$$0 \leq c_2 g(n) < f(n) < c_1 g(n)$$

Now, the above inequality cannot be true asymptotically as n becomes very large, $f(n)$ cannot be simultaneously greater than $c_2 g(n)$ and less than $c_1 g(n)$ for any constants $c_1, c_2 > 0$

Hence, no such $f(n)$ exists, i.e. the intersection is indeed the empty set.

USING LIMIT DEFINITIONS

We can use the limit definitions of $o(n)$ and $\omega(n)$ to draw same conclusion.

$$o(g(n)) = \lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$$

and

$$\omega(g(n)) = \lim_{n \to \infty} \frac{f(n)}{g(n)} = \infty$$

Both of these cannot hold true as n approaches ∞.

Practice Questions

1-14. *[5]* Prove by induction on $n \geq 1$ that for every $a \neq 1$,

$$\sum_{i=0}^{n} a^i = \frac{a^{n+1} - 1}{a - 1}$$

1-15. *[3]* Prove by induction that for $n \geq 1$,

$$\sum_{i=1}^{n} \frac{1}{i(i+1)} = \frac{n}{n+1}$$

Home Assignments

2.10 Exercises

Program Analysis

2-1. *[3]* What value is returned by the following function? Express your answer as a function of n. Give the worst-case running time using the Big Oh notation.

function mystery(n)
  $r := 0$
  for $i := 1$ *to* $n - 1$ *do*
    for $j := i + 1$ *to* n *do*
      for $k := 1$ *to* j *do*
        $r := r + 1$
  return(r)

2-2. *[3]* What value is returned by the following function? Express your answer as a function of n. Give the worst-case running time using Big Oh notation.

function pesky(n)
  $r := 0$
  for $i := 1$ *to* n *do*
    for $j := 1$ *to* i *do*
      for $k := j$ *to* $i + j$ *do*
        $r := r + 1$
  return(r)

2-3. *[5]* What value is returned by the following function? Express your answer as a function of n. Give the worst-case running time using Big Oh notation.

function prestiferous(n)
  $r := 0$
  for $i := 1$ *to* n *do*
    for $j := 1$ *to* i *do*
      for $k := j$ *to* $i + j$ *do*
        for $l := 1$ *to* $i + j - k$ *do*

Thank You