CmpE 300 - 1^{st} Assignment Solutions

Posted xx.11.2004

Question 1 (50 pts.): Prove or disprove:

$$O\left(\left(\frac{n^2}{\log\log n}\right)^{\frac{1}{2}}\right) = O(\lfloor\sqrt{n}\rfloor).$$

Answer: If the equality holds, then the following two conditions shall be met:

- $\exists\ n_0, c_0$ s.t.
$$\left(\frac{n^2}{\log\log n}\right)^{\frac{1}{2}} \leq c_0\lfloor\sqrt{n}\rfloor \tag{1}$$
for all $n \geq n_0$.
- $\exists\ n_1, c_1$ s.t.
$$\lfloor\sqrt{n}\rfloor \leq c_1\left(\frac{n^2}{\log\log n}\right)^{\frac{1}{2}} \tag{2}$$
for all $n \geq n_1$.

Let us look at the first of these:

$$\begin{aligned}\left(\frac{n^2}{\log\log n}\right)^{\frac{1}{2}} &\leq c_0\lfloor\sqrt{n}\rfloor \leq c_0(\sqrt{n}+1)\\ \frac{n^2}{\log\log n} &\leq c_0^2(n+2\sqrt{n}+1)\\ n^2 &\leq 4c_0^2 n\log\log n\end{aligned}$$

which obviously will not hold for all $n \geq n_0$, as the left hand side grows faster than the right hand side. Therefore, the equality is not valid.

Question 2 (50 pts.): What does the following function return (as a function of n)? What is the worst-case running time in big-O notation?

```
    function loops(n)
1.  m = 0
2.  for k = 1 to n do
3.      for l = 1 to k do
4.          for i = l to l + k do
5.              m = m + 1
6.          endfor
7.      endfor
8.  endfor
9.  return(m)
10. end
```

Answer: The value returned by the function is equal to the times the inner-loop is executed (line 5), as each such iteration increments m by 1. If we find how many times the loop is iterated, we will learn both the complexity bound and the value of m at the time of termination.

The outer loop (indexed by k) is run n times. For the j^{th} iteration, the second loop (indexed by l) is run j times. For each iteration of the second loop, the third loop (indexed by i) is executed $(l + k) - l + 1$ times, i.e. $j + 1$ times (because the value of k for the j^{th} iteration of the outer loop is j). Then the number of times the 5^{th} line is iterated is given by:

$$m = \sum_{j=1}^{n} j * (j + 1) = \frac{n(n + 1)(n + 2)}{3} \qquad (3)$$

Therefore the running time of the function can be expressed by $O(n^3)$ additions.