

## Lecture C9

### Response to 'Muddiest Part of the Lecture Cards'

(10 respondents)

1) *What is  $T(n)$ ?*

We are trying to get an equation of the time taken to solve a problem of size  $n$ . We represent the total time taken as  $T(n)$ . We will see more of this in detail in the next lecture.

2) *For  $T(n) = C_s n/2$ , the bound was  $n/2 \leq n$ , and we said that this was valid for  $n \geq 1$ , so  $n=0 = 1$ . But isn't  $n/2 \leq n$  valid for all  $n \geq 0$ ?*

Yes mathematically that is correct, but  $n=0$  is an empty array, and the worst case, best case and average case are all constant time.

3) *Why do we use Big-O?*

Big-O notation allows us to perform an asymptotic analysis on resource usage (the resource could be memory used or processor time). This allows us to compare two algorithms in terms of best case, worst case and average case behavior. If  $T(n)$  is the computation time of the algorithm, then its asymptotic behavior can be expressed as

$T(n) = O(f(n))$  such that  $T(n) \leq O(c f(n))$ , for all  $n \geq n_0$

The two constants  $c$  and  $n_0$  are defined such that the algorithm always executes in less than  $c \cdot f(n)$  time.

4) *What are constant time algorithms?*

Algorithms that always take a fixed amount of time irrespective of the size of inputs are called constant time algorithms. For example: inserting an element at the head of the linked list.

5) *Why is  $T(n) = c_1 n^2 + c_2 n = O(n^2)$ ?*

The computation of the constants is based on the definition of Big-O (see question 3 above):

$$\begin{aligned} T(n) &= c_1 n^2 + c_2 n \\ &\leq c_1 n^2 + c_2 n^2 \quad (\text{this is true for all } n \geq 1) \\ &\leq (c_1 + c_2) n^2 \end{aligned}$$

Hence  $c = c_1 + c_2$  and  $n_0 = 1$

6) *Why cant you compress  $O(N^2 \log P + N)$  to  $O(N^2 \log P)$ ?*

We are trying to find the upper bound on the execution time, when we have all terms of the same type, we choose the one with most significance and ignore lesser order terms. In the case of  $N^2 \log P + N$ , we do not know which the dominant term is. The  $N^2 \log P$  term dominates when  $N^2 \log P > N$ . Both terms contribute equally when  $P = 2^{1/n}$  (assuming the log base is 2), but since we do not know for sure what  $N$  and  $P$  are, we cannot eliminate the  $N$  term.

7) *No Mud*