Rick Boles

CS 3410: Data Structures

Homework 1 (Fall 2017) (Assignment 2 on Blazeview)

Due date: Monday, August 18, 2017 by 2:00 P.M.

1. After analysis, it is found that two programs (P1 and P2) have actual worst-case running times as:
T(P1) = 2500*N + log N and T(P2) = N2 + N.

Which program has the better guarantee on the running time if the input size is given by N ≤ 2000? Explain.[5 pts]

$2500*(2000) + \log(2000) = 5{,}000{,}003.301$ (used std log not $\log_2$)

worst case

$(2000)^2 + (2000) = 4{,}002{,}000$

$T(P2) = n^2 + n$ has the better run time because of the 2000 limit as shown above. With no limit, however $2500*n + \log n$ would be the best choice if input size could ever be infinite or even slightly larger than 2000.

2. Solving a problem requires running of an O(n2) step followed by two O(nlog n) steps in sequence. What is the total cost (running time) of solving the problem? Explain briefly.[5 pts]

So $n^2 + n\log n + n\log n$ or $n^2 \rightarrow n\log n \rightarrow n\log n$ will be $O(n^2)$ because n^2 is the greatest contributing factor

3. An algorithm takes 1 second for input size 1000. How large a problem (input size) can be solved in 1 minute if the running time is quadratic (assuming that low-order terms are negligible)? [5 pts]

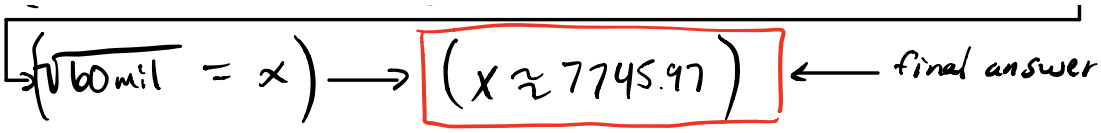
1st way I considered $1000 = 10^2$ and worked out for x:

$\left(\frac{1}{60} = \frac{1000}{x^2}\right) \rightarrow \left(\frac{1}{60} \cancel{=} \frac{1000}{x^2}\right) \rightarrow \left(x^2 = 60000\right)$

$\left(\sqrt{60{,}000} = x\right) \rightarrow \left(x \approx 244.95\right)$

however I believe the correct way is $\frac{1}{60} = \frac{1000^2}{x^2}$

$\left(\frac{1}{60} = \frac{1000^2}{x^2}\right) \rightarrow \left(\frac{1}{60} \cancel{=} \frac{1000^2}{x^2}\right) \rightarrow \left(x^2 = 60\text{mil}\right)$



4. For the following program fragment give a Big-O analysis of the running time. Briefly explain your answer.[5 pts]

```
int t = 0;
for(int i=1; i <= n; i++)
for(int j=1; j <= i*i; j++)
if(j % i == 0)
t++;
```

n
n*n (worst case) ← nested loop
c (least contributing factor)

$O(n^3)$

I worked this out mathmetically to help understand. for an example n=5; "i" will grow to "n". Since i*i worse case will be 5*5 or n*n. 25 or n^2 is nested by 5 and n respectively. This leaves me with n^3 or 125 which is (5*5*5) or (n*n*n). As I change values to other integers such as 6 or 7 the values stay equivilent to n. The key for me is that i becomes n in worst case scenario changing i*i to n*n giving me n*n*n or n^3.

Submission guideline: You have the following options to submit your assignment. Choose only one option that suits you best.

(i) Submit a handwritten copy in class. Writing should be clear (readable). Include Course number (CS3410), Semester (Fall 2017), Assignment number (Homework 1) and your name at the top. Failure to include these will not be accepted.
(ii) Scan your handwritten copy (including information mentioned in (i)) and submit it through BlazeVIEW dropbox. Scan resolution should be good (readable).
(iii) Type your answers on a document file using MS-Word. Include all information (mentioned in (i)) and submit through BlazeVIEW dropbox. Make sure symbols are appearing appropriately.