CS 202 – Data Structures (Spring 2007)

Homework Assignment 1: Algorithm Analysis

Assigned on 23 February, 2007 — Due by 23:00:00 on Thursday 15 March, 2007
Graded by (Mustafa) Tolga Eren (mustafae@...)

PLEASE NOTE:

- The deadline is hard: No exceptions will be made.
- Solutions must be your own: No cooperation is permitted.
- Late solutions will be penalised by 10 points for each day of delay, but solutions that are late by more than 2 days will get 0 points.
- Solutions must be submitted via the WebCT server (whose clock may differ from yours): No other method of submission will be accepted.

Problem 1 ($12 + 4 \cdot 1 = 16$ points, the 12 points being for a general reasoning)
A program for an algorithm takes 1 millisecond of CPU time on some computer for an input of size 64. How long do you predict it will take the same computer for an input of size 2048 if the running time of the algorithm (ignoring lower-order terms) is

a. $\Theta(\sqrt{n})$

b. quadratic

c. $\Theta(n^2 \log n)$

d. cubic

on an input of size n? Exhibit your reasoning; otherwise you get zero points for correct values.

Problem 2 ($12 + 8 \cdot 1 = 20$ points, the 12 points being for a general reasoning)
Complete the following table, which shows the sizes of a problem that can be solved in one hour of CPU time on some computer with an algorithm of the given complexity. So, for example, if the algorithm has $\Theta(\log n)$ complexity and you can solve a problem of size A in one CPU hour, what is the size of the problem that you can solve in one CPU hour if the CPU is 100 or 1000 times faster? Exhibit your reasoning; otherwise you get zero points for correct expressions.

Complexity	**Size when CPU Speed = 1**	**Size when CPU Speed = 100**	**Size when CPU Speed = 1000**
$\Theta(\log n)$	A		
$\Theta(\log^2 n)$	B		
$\Theta(2^n)$	C		
$\Theta(2^{n^2})$	D		

Problem 3 ($2 \cdot 20 = 40$ points)
The following procedure computes the maximum and the minimum of an array of n numbers:

```
void maxmin(int A[], int nBegin, int nEnd, int *pnMaxResult, int *pnMinResult)
{
     int nMax1, nMin1, nMax2, nMin2;
     int N = nEnd - nBegin + 1;
     if (N == 2) {
         if (A[nBegin] < A[nEnd]) {
            *pnMaxResult = A[nEnd];
            *pnMinResult = A[nBegin];
         }
         else {
            *pnMaxResult = A[nBegin];
            *pnMinResult = A[nEnd];
         }
         return;
     }
     maxmin(A, nBegin, nBegin+N/2 - 1, &nMax1, &nMin1);
     maxmin(A, nBegin+N/2, nEnd , &nMax2, &nMin2);
     if (nMax1 < nMax2) {
            *pnMaxResult = nMax2;
         }
         else {
            *pnMaxResult = nMax1;
         }
     if (nMin1 < nMin2) {
            *pnMinResult = nMin1;
         }
         else {
            *pnMinResult = nMin2;
         }
     return;
 }
```

Assume that n is a power of 2 and that this procedure is called from a main program such as:

```
main()
{
  int A[]={1,3,5,7,2,4,6,8};
  int nMax,nMin;
  maxmin(A,0,7,&nMax,&nMin);
  printf("Max %d  Min %d\n", nMax,nMin);
}
```

Answer the following questions:

a. Write a recurrence that computes the number of comparisons involving array elements or their maxima and minima as a function of the array size n. So a comparison like `if (N == 2) ...` should *not* be counted. Be very careful about the initial conditions.

b. Solve the recurrence exactly by successive expansion.

Exhibit your reasoning; otherwise you get zero points for correct answers.

Problem 4 ($4 \cdot 6 = 24$ points)
Compute values of n_0 and c that can be used to prove the following relationships:

a. $25n^4 - 19n^3 + 13n^2 - 106n + 77 = O(n^4)$

b. $2^{n+10} = O(2^n)$

c. $n^{10} = O(3^n)$

d. $\log n = O(\sqrt{n})$

Exhibit your reasoning; otherwise you get zero points for correct values.

What and How to Submit When?

Your solution must be in a single PDF file. Take our warning on plagiarism **very seriously**. We assume that by submitting your solution, you are certifying that you are submitting your own work. Take the following steps:

1. Name the file as *LastnameName-StudentID-hw1.pdf*. Do not use any special Turkish characters in the file name.

2. Compress your file, giving *MehmetogluAli-15432-hw1.zip* for example. Make sure it uncompresses properly, reproducing your file exactly, and corresponds to assignment 1.

3. Submit this compressed file via WebCT by the deadline given on the first page.

For late submissions, we will first grade your solution and then discount accordingly. So, assuming you got 90 points, if you were late at least one second but at most one day, then your actual grade will be 80 points; if you were late more than one day but at most two days, then your actual grade will be 70 points; if you were late more than two days, then your actual grade will be 0 points and your solution will not be graded.

Have fun!