Homework 1: Algorithm Analysis

Handed out Thursday, Jan 26. Due at the start of class Thursday, February 1.

Problem 1. Characterize using the Big-Oh notation the worst case running times of the following algorithms. Show how you derived your answer.

(a) Let A be a given array of n integers.
for $i \leftarrow 0$ to n **do**
  for $j \leftarrow i$ to $4i$ **do**
    for $k \leftarrow 0$ to j **do**
      Let $A[i] \leftarrow A[i] + A[j] + A[k]$.
    end for
  end for
end for

(b) Let A be a given array of n integers.
for $i \leftarrow 0$ to n **do**
  for $j \leftarrow 0$ to $2i$ **do**
    for $k \leftarrow 0$ to 2^j **do**
      Let $A[i] \leftarrow A[i] + A[j] + A[k]$.
    end for
  end for
end for

(c) **for** $i \leftarrow 0$ to n **do**
  $j = i;$
  while $j > 0$ **do**
    Let $A[i] \leftarrow A[i] + A[j]$.
    $j / = 2;$
  end while
end for

Problem 2. For each part, rank the following functions in increasing order of asymptotic growth rate. For functions that are equivalent, group them together.

(a) $200n^2, 5n^3 + \log n, 10n + 8/n^5$

(b) $4\sqrt{n} + 4\lg(n^3), 5\log_4 n, 5\lg^2 n + 10\lg n$

(c) $2^{(3\lg n)}, 500n^2, 2^{\sqrt{n}}$

(d) $3n\lg\lg n, n\lg^6 n + 8n^2$

Problem 3. Many numerical applications use square matrices that are *lower triangular*, that is, all entries lying strictly above the main diagonal are zero. An *upper triangular matrix* is defined similarly. Suppose that we want to implement a data structure to store such a matrix using the minimum space. To do this we store the entries in a one-dimensional array, and use an indexing function to map a row-column pair $[i, j]$ to an integer offset in this array.

i \ j	0	1	2	3	4
0	0				
1	1	2			
2	3	4	5		
3	6	7	8	9	
4	10	11	12	13	14

i \ j	0	1	2	3	4
0	0	1	2	3	4
1		5	6	7	8
2			9	10	11
3				12	13
4					14

Figure 1: Lower- and upper-triangular array offsets for $n = 5$.

(a) As a function of n, derive the number of nonzero entries that an $n \times n$ lower (or upper) triangular matrix might have.

(b) Given an $n \times n$ lower triangular matrix, give the function that maps an index pair $[i, j]$ to an offset into a 1-dimensional array, as shown in the figure above left. Your function should run in $O(1)$ time.

(c) Repeat part(b) for an upper triangular matrix using the offset method shown in the right figure.

Problem 4. Use formal definitions (not the Limit Rule) to establish that $4n^3 - 3n + 24 \in \Theta(n^3)$. Specific values of the constants you used to satisfy the conditions, and show how you arrived at these values.