CSE 6140 / CX 4140
Computational Science & Engineering (CSE) Algorithms

Homework 1

Please type in all answers.

1. Suppose you have five algorithms that have the following running times. (Assume they are the exact running times). n denotes input size.

n^2

n^3

$100\,n^2$

$n\,log\,n$

2^n

Consider three different input sizes 10, 100, 1000.

a. (5 points) What is the running time of each algorithm for each input size? Use a calculator and provide your answers in scientific notation; i.e. write $10,000$ as 10^4.

b. (5 points) The running time of another algorithm is n!. Consider input sizes of 10, 50 and 100. What is the running time of the algorithm for each input size?

2. (10 points) Consider a for loop that runs for n steps. There's a function $g(\cdot)$ inside that for loop. How would you describe the running time of the loop (in O-notation) if $g(\cdot)$ completes in:

n steps

$log\ n$ steps

k steps, where k is a constant

3. The worst-case running time of five algorithms are given below.

$O(n^{2.5})$

$O(\sqrt{2n})$

$O(n+10)$

$O(10^n)$

$O(100^n)$

a. (2 points) Which running time grows the fastest with increasing n?

b. (2 points) Which running time grows the slowest with increasing n?

c. (2 points) Arrange the algorithms in ascending order of running time.

d. (2 points) In your ascending order, where would you place an algorithm that has worst-case running time of $O(n!)$?

e. (2 points) Where would you place an algorithm that has worst-case running time of $O(n^2 \log n)$ time?

4. (10 points) Is $2^{n+1} = O(2^n)$? Is $2^{2n} = O(2^n)$? Use the formal definition of O-notation to justify your answer?

5. (5 points) Express the function $n^3/1000 + 100n^2 - 100n + 3$ in terms of Θ-notation.

6. (15 points) Consider sorting n numbers stored in array $A[1 : n]$ by first finding the smallest element of $A[1 : n]$ and exchanging it with the element in $A[1]$. Then find the smallest element of $A[2 : n]$ and exchange it with $A[2]$. Then find the smallest element of $A[3 : n]$, and exchange it with $A[3]$. Continue in this manner for the first $n-1$ elements of A. Write pseudocode for this algorithm, which is known as **selection sort**. Why does it need to run for only the first $n-1$ elements, rather than for all n elements? Give the worst-case running time of selection sort in Θ-notation. Is the best-case running time any better?

7. Let $A[1 : n]$ be an array of n distinct numbers. If $i < j$ and $A[i] > A[j]$, then the pair (i, j) is called an **inversion** of A.

a. (5 points) List the five inversion of the array $\{2, 3, 8, 6, 1\}$.

b. (5 points) What array with elements from the set $\{1, 2, 3, ..., n\}$ has the most inversions? How many does it have?

c. (10 points) Give an algorithm that determines the number of inversions in an array of n elements in $O(n \log n)$ worst-case time. (*Hint:* Modify merge sort).

8. (20 points) You are a robber planning to rob houses along a street. You know that each house has a certain amount of money stashed. To avoid raising alarms, you decide you won't rob adjacent houses. Which houses should you rob to get maximum cash? Provide the pseudocode.