CSC345 Discussion 5

Project 2, Going over the quiz, Big O Practice

Project 2 is up

Demo

Any Questions?

Quiz Question 1

Suppose that a particular algorithm has time complexity $T(n) = 3 * 2^n$ and that executing an implementation of it on a particular machine takes t seconds for n inputs. Now suppose that we are presented with a machine that is 64 times as fast. How many inputs could we process on the new machine in t seconds?

Quiz Solution 1

Let n' be the new number of inputs, so we have

$\frac{3*2^{n'}}{64} = 3 * 2^n$

$2^{n'=64*2^n}$

$\log_2 2^{n'} = \log_2 64 * 2^n$

$n' = \log_2 64 * 2^n$

$n' = \log_2 64 + \log_2 2^n$

$n' = 6 + n$

Quiz Question 2

Circle all the true relationships for the following functions

$f(n) = 2^n$

$g(n) = n \log n$

a. $f(n) \in o(g(n)$

b. $f(n) \in O(g(n)$

c. $f(n) \in \Theta(g(n)$

d. $f(n) \in \Omega(g(n)$

e. $f(n) \in \omega(g(n)$

Quiz 2 Solution

Keep in mind the defntion of Ω and ω

Ω

For $T(n)$ a non-negatively valued function, $T(n)$ is in the set $O(f(n))$ if there exists two positive constants c and n_0 such that $T(n) \leq cf(n)$ for all $n > n_0$

ω

For $T(n)$ a non-negatively valued function, $T(n)$ is in the set $O(f(n))$ if there exists two positive constants c and n_0 such that $T(n) < cf(n)$ for all $n > n_0$

Quiz 2 Solution

Solution d and e

Quiz Question 3

The worst case for sequential search occurs when the last element of the array is the value being searched for

a. True
b. False

Quiz 3 Solution

True

Quiz Question 4

The best case for the sequential search algorithm occurs when the array has only a single element.

a. True
b. False

Quiz 4 Solution

False

Best case cost refers to a best problem instance AS THE INPUT GETS BIG.

So it makes no sense to talk about a growth rate or a best case of a fixed input size.

The best case for sequential search occurs when the first element of the array (for whatever size) is the value being searched for.

Exercise 3.11

For each of the following pairs of functions, either $f(n)$ is in $O(g(n))$, $f(n)$ is in $\Omega(g(n))$, or $f(n) = \Theta(g(n))$. For each pair, determine which relationship is correct. Justify your answer, using the method of limits discussed in Section 3.4.5.

(c) $f(n) = \log^2 n; \quad g(n) = \log n.$

(d) $f(n) = n; \quad g(n) = log^2 n.$

Exercise 3.12

Determine Θ for the following code fragments in the average case. Assume that all variables are of type **int**.

(a)
```
a = b + c;
d = a + e;
```

(b)
```
sum = 0;
for (i=0; i<3; i++)
   for (j=0; j<n; j++)
      sum++;
```

(d)
```
for (i=0; i < n-1; i++)
   for (j=i+1; j < n; j++) {
      tmp = AA[i][j];
      AA[i][j] = AA[j][i];
      AA[j][i] = tmp;
   }
```

(i)
```
sum = 0;
if (EVEN(n))
  for (i=0; i<n; i++)
    sum++;
else
  sum = sum + n;
```

Exercise 3.19

Modify the binary search routine to support search in an array of infinite size. In particular, you are given as input a sorted array and a key value K to search for. Call n the position of the smallest value in the array that is equal to or larger than X. Provide an algorithm that can determine n in $\mathrm{O}(\log n)$ comparisons in the worst case. Explain why your algorithm meets the required time bound.

Exercise 3.24

Prove that if an algorithm is $\Theta(f(n))$ in the average case, then it is $O(f(n))$ in the best case.

Exercise 4.11

Use the space equation of Section 4.1.3 to determine the break-even point for an array-based list and linked list implementation for lists when the sizes for the data field, a pointer, and the array-based list's array are as specified. State when the linked list needs less space than the array.

(a) The data field is eight bytes, a pointer is four bytes, and the array holds twenty elements.

(b) The data field is two bytes, a pointer is four bytes, and the array holds thirty elements.