Data Structures and Algorithms CSE220

Prof. Ramesh Ragala

September 4, 2015

Question - 1

Given 'N' objects, which are coloured as red, white and blue. Sort these objects so that objects of the same colour are adjacent, with the colours in the order red, white and blue. Design an algorithm with a time complexity of O(nlog n). **10M**

KEY

We can use 3-way partition Quick Sort to solve the above problem. The problem was posed with three colours, here '0', '1' and '2'. The array is divided into four sections:

a[1..Low-1] zeroes (red)
a[Low..Mid] ones (white)
a[Mid..Hi] unknown
a[Hi+1..N] twos (blue)

Three-way partitioning logic: assumes zero-based array indexing. It uses three indices i, j and n, maintaining the invariant that i$\leq$ j.

Algorithm threewaypartition(a,mid)

```
{
i ← 0;
j ← 0;
n ← Len(a − 1);
while j ≤ n do
    if a[i] < mid then
        swapA[i]andA[j]
        i ← i + 1
        j ← j + 1
    else if a[j] > mid then
        SwapA[j]andA[n]
        n ← n − 1
    else
        j ← j − 1
    end if
```

Question - 2

Let a_n denote number of bit strings of length-n (n≥3) that do not have two consecutive 1s ("valid strings"). Find a recurrence relation with initial conditions for the sequence $\{a_n\}$ and also solve the recurrence relation. **10M**

KEY

The number of valid strings is the number of valid strings ending with a 0 plus the number of valid strings ending with a 1.
That gives us the following recurrence relation:$a_n = a_{n-1} + a_{n-2}$
The Initial Conditions are: $a_1 = 2$ (0 and 1), $a_2 = 3$ (01, 10 and 11),
$a_3 = a_2 + a_1 = 3 + 2 = 5$, $a_4 = a_3 + a_2 = 5 + 3 = 8$
This sequence satisfies the same recurrence relation as the Fibonacci sequence. Since $a_1 = f_3$ and $a_2 = f_4$, we have $a_n = f_{n+2}$.

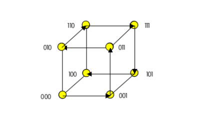
Question - 3

n-dimensional Hyper Cube (H_n) has 2^n nodes with each node connected to n-other nodes. Each node is represented with bit string of size-n. The bit strings of adjacent nodes are differ with one bit only. Hamilton path is a sequence of edges that visit each node exactly once. Simulate the Hamilton path on n-dimensional Hyper Cube using Towers of Hanoi and write the path **10M**

Key

1. Algorihtm for Towers of Hanoi.

Simulation of TOH with n-hypercube:1.Initialize to (0,..,0) and flip the ith coordinate/node/vertex when the ith disk/code is moved.



Question - 4

Given a sorted list L, design an efficient iterative/recursive algorithm that searches an element of all pairs whose sum S is exactly equal to the given value x. Analyse the time complexity of the algorithm

KEY

Suppose we have a set $\{10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110\}$
Assume Search Element is 50 $\rightarrow$ probably $\{(10, 40), (20, 30)\}$
Procedure: calculate the difference between search element or key with first(second, third ...). Now identify the differed values is in the list or not using binary search. So write the algorithm for this procedure

Question - 5a

List the following functions according to their order of growth from the **lowest to the highest**:
(n-2)!, 5 $\log(n+100)^{10}$, 2^{2n}, $0.001n^4+3n^3+1$, $\log^2 n$, $\sqrt[3]{n}$ and 3^n **1M**

Key

$\log^2 n$, $\sqrt[3]{n}$, 5 $\log(n+100)^{10}$, $0.001n^4+3n^3+1$, 3^n, 2^{2n}, and (n-2)!

QUESTION - 5B

ALGORITHM Mystery(n)
//Input: A nonnegative integer n
begin
$S \leftarrow 0$;
for $i \leftarrow 1$ to n do
{
$S \leftarrow S + i * i$
}
return S
end

i. What does this algorithm compute?
ii. What is its basic operation?
iii. How many times is the basic operation executed?
iv. What is the efficiency class of this algorithm?

KEY - 5B

i. compute the squares of the numbers
ii. Multiplication
iii. n times
iv. O(n)

Question - 5c

Can the master method be applied to the recurrence T(n) = 4T(n/2)+n^2 log n? Why or why not? Give an asymptotic upper bound for this recurrence. **5M**

Key-5c

Master Theorem is not applied to this recurrence relation, even it looks like the master theorem general format. Due to f(n)=n^2 log n. Because f(n) can be obey the Case-1, Case-2 and Case-3 of master theorem. so we have to use substitution method. guess is $\Theta(n^2 log^2 n)$. So we have to prove it using substitution method.