DAA - Tutorial 2

Name - Tisha Aggarwal
Sec - CST SPL 2
Roll No - 31

Q1) for i loop, $i = 1$ =
$= 1+2$
$= 1+2+3$
$= 1+2+3+ \dots + n$

for j loop, $j = 1$
$= 2$
$= 3 \dots$ n times

$\therefore 1+2+3+ \dots n < n$

$\frac{n(n+1)}{2} < n$

$n^2 + n < 2n$

$n^2 < n$

$n < \sqrt{n}$ or $n \simeq \sqrt{n}$

$\therefore \sum_{i=1}^{m} (1) \Rightarrow 1+1+1+ \dots \sqrt{n}$ times

$T(n) = O(\sqrt{n})$. (Ans)

Q2) Recurrence relation :-

$fib(n) = fib(n-1) + fib(n-2)$

fib(n) : if n <= 1
return 1
return fib(n-1) + fib(n-2)

Time Complexity:

$T(0) = 1$

$T(n) = T(n-1) + T(n-2) + c$
$= 2T(n-2) + c$

$T(n-2) = 2^* (2T(n-2-2) + c) + c$
$= 2^* (2T(n-4) + c) + c$

Tisha

2

$= 4T(n-4) + 3c$

$T(n-4) = 2*(4T(n-4)+3c)+c$

$= 2^K * T(n-K) + (2^K - 1)c$

$\Rightarrow n - K = 0$

$n = K$

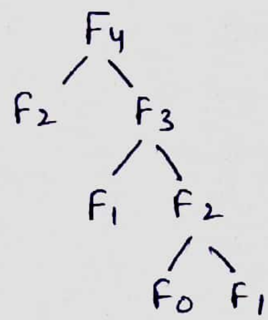
$T(n) = 2^n * T(0) + (2^n - 1)c$

$= 2^n * 1 + 2^n c - c$

$= 2^n (1+c) - c$ (Constanst ignored)

$= O(2^n)$

Space Complexity :- Space is proportional to max. depth of the recursive tree.



Hence the space complexity of Fibonacci recursive is O(N).

Q3) 1- O(n logn) — Quick Sort

```
void Quicksort (int arr[], int l, int h)
{
  if (l<h)
  {
  int p = divide(arr, l, h);
   Quicksort (arr, l, p-1);
   Quicksort (arr, p+1, h);
  }
}
int divide (int arr[], int l, int h)
 {
   int pivot = arr[h];
   int i = (l-1);
   for (int j = l; j<=h-1; j++)
   {
     if (arr[i] < pivot)
     {
       i++;
       swap (&arr[i], &arr[j]);
     }
   }
   swap (& arr[i+1], &arr [high]);
   return (i+1);
 }
```

2 - n^3

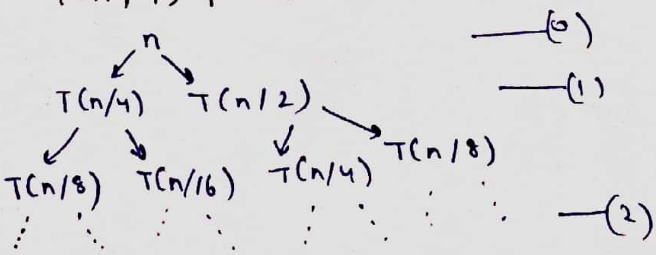
```
for (i=0; i<n; i++)
{ for (j=0; j<n; j++)
  { for (int k=0; k<n; k++)
    {
      printf("x");
    }
  } }
  return 0;
}
```

3 - O(log (log n))

```
int countP(int n)
{  if (n < 2)
   return 0;

  boolean[] nonP = new boolean[n];
     nonP[1] = true;
   int numNonP = 1;
   for (int i = 2; i<n; i++)        // O(n)
  {  if (nonP[i])
       continue;
    int j = i * 2;
    while (j<n)             // O(log(log(n)))
    { if (!nonP[j])
      {
        nonP[j] = true;
         numNonP++;
      }
      j+ = i;
    }
 } }
   return (n-1) - numNonP;
```

Q4) T(n) = T(n/4) + T(n/2) + cn^2



Scanned with CamScanner

(4)

$0 \to cn^2$

$1 \to \frac{n^2}{4^2} + \frac{n^2}{2^2} = \frac{c5n^2}{16}$

$2 \to \frac{n^2}{8^2} + \frac{n^2}{16^2} + \frac{n^2}{4^2} + \frac{n^2}{8^2}$

$= \left(\frac{5}{16}\right)^2 n^2 c.$

Maximum level $= \frac{n}{2^k} = 1$

$\Rightarrow K = \log_2 n$

$\therefore T(n) = c(n^2 + (5/16)n^2 + (5/16)^2 n^2 + \ldots)$

$= c((5/n)^{\log n} n^2)$

$T(n) = cn^2 \left[(1) + (5/16)n^2 + (5/16)n^2 + \ldots + \left(\frac{5}{16}\right)^{\log n}\right]$

$= cn^2 \left(\frac{1-(5/16)^{\log n}}{1-5/16}\right)$

$= cn^2 \left(\frac{11}{5}\right)\left(1-(5/16)^{\log n}\right)$

$= O(\cancel{c}n^2)$

$= O(n^2)$ (Ans)

Q5) ~~for~~ for $i = 1$ $\quad j = 1$ $\quad j = (n-1)/i$ times

2 $\quad$ 1+3+5

3 $\quad$ 1+4+7

$\vdots$ $\quad \vdots$

n

$\Rightarrow \sum_{i=1}^{n} \frac{(n-1)}{i} \Rightarrow T(n) = \frac{(n-1)}{1} + \frac{(n-1)}{2} + \ldots + \frac{(n-1)}{n}$

$\Rightarrow T(n) = n\left[1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{n}\right]$

$= n\log n$

$= O(n\log n)$

tush

Scanned with CamScanner

Q6) for (i=2; i<=n; i=pow(i,k))

{

//o(n)

}

pow (i,k)

$i = 2^1$ $\quad\therefore 2^{k^m} <= n$

$= 2^k$ $\quad k^m = \log_2 n$

$= 2^{k^2}$ $\quad m = \log_k \log_2 n$

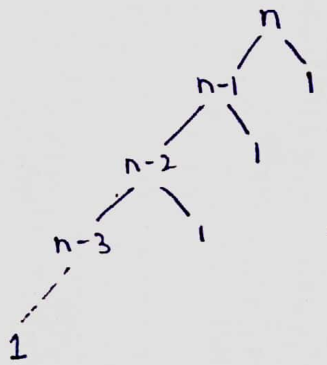
$= 2^{k^3}$

and $O(1) \rightarrow$ m times

$\therefore T(n) = O(\log_k \log_2 n) + \cancel{O(1)}$

$= O(\log_k \log_2 n)$

Q7) $T(n) = T(n-1) + O(1)$



'n' levels

$\rightarrow$ difference in height $= n-2$

$\rightarrow$ lowest height $=2$

highest height $=n$

$T(n) = [T(n-1) + T(n-2) + \ldots + T(1) + O(1)] \times n$

$= n \times n$

$= O(n^2)$.

Q8) a) $100 < \log(\log n) < \log n < (\log n)^2$

$< \sqrt{n} < n < n(\log n) < \log(n!)$

$< n^2 < 2^n < 4^n < 2^{2^n}$

b) $1 < \log(\log n) < \sqrt{\log(n)} < \log n$

$< \log 2n < 2(\log n) < n < n(\log n) < 2n < 4n$

$< \log(n!) < n^2 < n! < 2^{2^n}$

(c) $96 < \log_8 n < \log 2n < 5n < n(\log_6 n) < n(\log_2 n) < \log(n!) < 8n^2$

$< 7n^3 < n! < 8^{2n}$

Scanned with CamScanner