Rohan Saini

CST

15

① for $j=1$ $\quad i=1$

$j=2$ $\quad i=1+2=3$

$j=3$ $\quad i=1+2+3=6$

$j=4$ $\quad i=1+2+3+4=10$

$1+2+3+4+\cdots$ m times $< n$

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

$$m \approx \sqrt{n}$$

$$\Rightarrow T(n) = O(\sqrt{n})$$

②

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

$T(n) \approx 2T(n-1) + 1$ —— ①

put $n = n-1$ in eqn ①

$T(n-1) \approx 2T(n-2) + 1$ —— ②

put ② into ①

$T(n) \approx 2[2T(n-2)+1] + 1$

$T(n) \approx 4T(n-2) + 2 + 1$ —— ③

put $n = n-2$ into ①

$T(n-2) \approx 2T(n-3) + 1$ —— ④

put ④ into ③

$T(n) \approx 4[2T(n-3)+1] + 2 + 1$

$T(n) \approx 8T(n-3) + 4 + 2 + 1$

$\Rightarrow T(n) = 2^K T(n-K) + 2^{K-1}$

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

let $n-K = 1$

$= 2^K (1) \Rightarrow 2^K$

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

(3) ij n(logn)

```
void mergesort (int arr[], int l, int r)
{ if (l < r)
  { int mid = l + (r-l)/2
    mergesort (arr, l, mid);
    mergesort (arr, mid+1, r);
    merge (arr, l, mid, r) }
}

void merge(int arr[], int l, int mid, int r)
{ ~~int t[r+1];~~
  int t[r+1];
  int i=l, j= mid+1; K=0;
  while (i<= mid && j<=r)
  { if (arr[i] > arr[j])
      t[K++] = arr[j++]
    else t[K++] = arr[i++];
  }
  while (i<= mid)
      t[K++] = arr[i++];
  while (j<= r
      t[K++] = arr[j++];
  for(int i=0; i<r+1; i++)
   { arr[i] = t[i]; }
}
```

(3) (ii) n^3

```
for(int i=0; i<n; i++)
{
    for(int j=0; j<n; j++)
    {
        for(int k=0; k<n; k++)
            count++;
    }
}
```

(3) (iii) log(log n)

```
for(i=2; i<n; i=i*i)
    c++;
```

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

$T(n) = 1 \cdot T(n/2) + cn^2$

master's method

$a = 1, b = 2, f(n) = cn^2$

$c = \log_2 1 = 0$

$f(n) > (n)^0$

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

(5) $T(n) = \sum_{i=1}^{n} \sum_{j=1}^{n-1} O(1)$

$= \sum_{i=1}^{n} (n-1)$

$= n(n-1)$

$= n^2 - n$

$= O(n^2)$ Ans

(6) $i = 2^1, 2^k, 2^{k^2}, 2^{k^3} \ldots n$

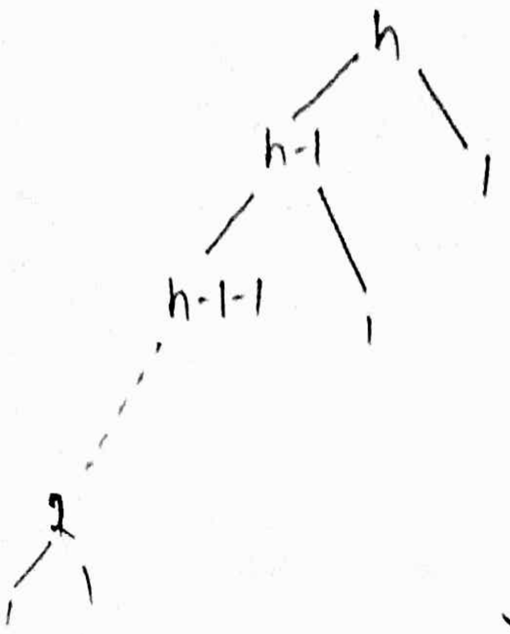
$a_n = 2^{k^m} <= n$

$k^m <= \log n$

$m < \log_k \log n$

$\sum_{i=1}^{m} 1+1+1+ \cdots m \text{ times} = O(\log_k \log_2 n)$

7 $T(n) = T(n-1) + O(1)$



$T(n) = (T(n-1) + T(n-2) + \cdots T(1))$

$(1+1+1+1 + \cdots n \text{ times})$

$T(n) = n \times n$

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

lowest level = 2

highest level = n

diff = n - 2

8

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) $1 < \log\log n < \sqrt{\log n} < \log n < \log 2n < 2\log n < n < n\log n < 2n < 4n < \log(n!) < n^2 < n < 2\log 2n < 5n$

c) $96 < \log_8 n < \log_2 n < 5n < n\log_6 n < n\log_2 n < \log(n!) < 5n^2 < 7n^3 < n! < 8^{2n}$