Name: Muhammad Mustafa Kamal Malik
Roll # 24l6l0047

Q1. Through Divide & conquer

A[1 — n]

```
foo(inp, s, e)
    if (s == e)
        return inp[s]
    else
     mid = floor((s+e)/2)
    left = foo(inp, s, mid)
    right = foo(inp, mid+1, e)
    if left → Right
            return left
    return right
```

Q2. $T(n) = T(n/2) + T(n/2)$

$= 2T(n/2) + 1$

Iter 1,

$T(n/2) = 2T(n/4) + 1$

$= 4T(n/4) + 1 + 1$

Iter 2,

$T(n/4) = 2T(n/8)$

$T(n) = 2(4T(n/8))$

$= 8T(n/8) + 1 + 1$

iter 3,

$T(n/8) = 2T(n/16)$

$T(n) = 2T(8T(n/16))$

$= 16T(n/16) + 1 + 1$

General form : $T(n) = 2^k T(n/2^k) + k$

$2^k = n$

$k = \log_2 n$

$$T(n) = nT(n/n) + n\log_2 n$$
$$= n + n\log_2 n$$
$$= O(n\log_2 n)$$

Q3. func (1) $(n) = O(n^{4/3})$

func 2 (int n) $= O(n^4)$

func 3 (int n, int m) $= O(n^{1/2}) \times O(\log^{3/2} m)$

$$= n^{1/2}\log^{3/2} m$$

function 4 $= O(n^{2/2})$

$$= O(\log^{\sqrt{n}} \cdot n^{1/5})$$

$$= O(\log_{2^{.5}} n)$$

$$= n^{13/3} + n^4 + n^{1/2}\log^{3/2} m + n^{2/2}(\log^{\sqrt{n}} n^{1/5})(\log_{2^{.5}} n)$$

Q5.

It's not possible to apply divide & conquer strategy effectively to solve the problem of concatenating n copies of a string. A simple loop is enough to solve the problem with a time complexity of O(n). However, we can use other techniques like StringBuilder / StringBuffer to optimize the performance.

Q4. Dry Run

Initial array → [39, 11, 77, 9, 17, 55, 27, 36, 22, 33, 22, 47]

Build Max heap → [77, 47, 55, 36, 22, 33, 27, 9, 27, 11, 39]

Extracting,

→ 77	→ 39
→ 47	→ 37
→ 36	→ 27
→ 33	→ 11
→ 55	→ 9
→ 22	

final array →

[9, 11, 22, 27, 36, 39, 55, 33, 36, 47, 55, 77]