MTH-681 Analysis and Design of Algorithms
SECOND MIDTERM EXAM

Mostafa Hassanein

22 JUNE 2025

Question 1

1.

$$
\begin{aligned}
& 2^{n+1} = 2 * 2^n < 3 * 2^n \\
\implies & \exists c = 3, n_0 = 0, \text{such that } \forall n > n_0 : 2^{n+1} < c * 2^n \\
\implies & 2^{n+1} = O(2^n).
\end{aligned}
$$

2.

$$
\begin{aligned}
& 2^{2n} = (2^n)^2 = 2^n * 2^n \\
\implies & \nexists c, n_0, \text{such that } \forall n > n_0 : 2^{2n} < c * 2^n \\
\implies & 2^{2n} = \Omega(2^n) \\
\implies & 2^{2n} \neq O(2^n).
\end{aligned}
$$

3.

We can obtain an upper-bound on the summation using an integral bound:

$$
\begin{aligned}
\sum_{k=1}^{n} k \ln k &\leq \int_1^{n+1} x \ln x dx \\
&= \left[\frac{1}{2} x^2 \ln x - \frac{1}{4} x^2 \right]_{x=1}^{n+1} \\
&= \frac{1}{2}(n+1)^2 \ln (n+1) - \frac{1}{4}(n+1)^2 + c \\
&\leq \frac{1}{2}(n+1)^2 \ln (n+1).
\end{aligned}
$$

Next, we use the limit test:

$$
\begin{aligned}
L &= \lim_{n \to \infty} \frac{\frac{1}{2}(n+1)^2 \ln (n+1)}{n^2 \ln n} \\
&= \lim_{n \to \infty} \frac{\frac{1}{2}(n^2+2n+1) \ln (n+1)}{n^2 \ln n} \\
&= \lim_{n \to \infty} \frac{\frac{1}{2} n^2 \ln (n+1) + n \ln (n+1) + \frac{1}{2} \ln (n+1)}{n^2 \ln n} \\
&= \lim_{n \to \infty} \frac{\frac{1}{2} n^2 \ln (n+1)}{n^2 \ln n} \\
&= \lim_{n \to \infty} \frac{\frac{1}{2} n^2 \ln n + \ln 1}{n^2 \ln n} \\
&= \lim_{n \to \infty} \frac{\frac{1}{2} n^2 \ln n}{n^2 \ln n} = \frac{1}{2}.
\end{aligned}
$$

$\implies \sum_{k=1}^{n} k \ln k = \Theta(n^2 \ln n) \implies \sum_{k=1}^{n} k \ln k = O(n^2 \ln n).$

Question 2

1.

Scrable[S, 1, 2]:
mid = 1
After 2 calls to merge sort help S won't change: S = [5,2]
The first call to Scrambe(S, 1, 1), results in: S[1] = S[1] * S[2] = 5*2 = 10.
The second and third calls to Scrable won't do anything.
The fourth call to scrable, results in: S[1] = S[1] * S[2] = 10*2 = 20.
The final calls to merge sort help won't do anything.
Therefore, the final result is: S = [20, 2]

2.

$$\begin{aligned} T(n) &= 4T(\frac{n}{2}) + 4\Theta(\frac{n}{2}\lg\frac{n}{2}) \\ &= 4T(\frac{n}{2}) + \Theta(n\lg n). \end{aligned}$$

3.

$$\begin{aligned} a &= 4 \\ b &= 2 \\ \lg_b a &= \lg_2 4 = 2 \\ n^{\lg_b a} &= n^2 \\ f(n) &= n\lg n \end{aligned}$$

$$\begin{aligned} &\implies f(n) = O(n^{2-\epsilon}), \text{ for } \epsilon = 0.5 \\ &\implies \text{First Case} \\ &\implies T(n) = \Theta(n^2). \end{aligned}$$

Question 3

Algorithm:

```
  Sequence LCS(s1, s2) {
  return LCSRecursive(s1, 1, m, s2, 1, n, 0)
}


Sequence LCSRecursive(Sequence s1, Index low_1, Index high_1, Sequence
    s2, Index low_2, Index high_2, Sequence partialResult) {
  if (low_1 > high_1 || low_2 > high_2) {
    return partialResult;
  }
  else {
    if (s[low_1] == s[low_2]) {
      LCSRecursive(s1, low_1+1, high_1, s2, low_2+1, high_2,
          partialResult + 1);
    }
    else if (s[low_1] < s[low_2]) {
      LCSRecursive(s1, low_1+1, high_1, s2, low_2, high_2,
          partialResult);
    }
    else {
      LCSRecursive(s1, low_1, high_1, s2, low_2+1, high_2,
          partialResult);
    }
  }
}
```