MTH-681 Analysis and Design of Algorithms
MIDTERM EXAM

Mostafa Hassanein

8 May 2025

Question 1

Theorem 1. *$W_t(n) = T(n) = \Theta(n \lg n)$.*

Proof.

In the worst case of this algorithm the variable "$found$" is never set to true (because on each iteration the search item is not found in the sequence S) and thus the while loop executes $n - 1 = \Theta(n)$ times.

On each iteration we call BSearchHelp on an input of size i.

Since BSearchHelp is an $\Theta(\lg n)$ algorithm, then on each iteration we incur $\Theta(\lg i)$.

The total time is then given by the summation:

$$\begin{aligned} T(n) &= \sum_{i=2}^{n} \Theta(\lg i) \\ &= \sum_{i=2}^{n} c_1 \lg i \\ &= c_1 \sum_{i=2}^{n} \lg i \end{aligned}$$

<u>Upper Bound:</u>

$$\begin{aligned} T(n) &= c_1 \sum_{i=2}^{n} \lg i \\ &\leq c_1 \sum_{i=2}^{n} \lg n \\ &\leq c_1 n \lg n \end{aligned}$$

Take $c_1 = 1$ and $n_0 = 1$, then:
$\implies T(n) = O(n \ln n)$.

<u>Lower Bound:</u>

□

Question 2

$$\begin{aligned} a &= 2 \\ b &= 8 \\ \log_b a &= \log_8 2 = 3 \\ n^{\log_b a} &= n^3 \end{aligned}$$

1.

$$\begin{aligned} & f(n) = n \\ \implies & f(n) = O(n^2) \\ \implies & f(n) = O(n^{\log_b a - \epsilon}) \quad \text{for } \epsilon = 1 \\ \implies & \text{Case 1} \end{aligned}$$

2.

$$\begin{aligned} & f(n) = n^3 \\ \implies & f(n) = \Theta(n^3) \\ \implies & f(n) = \Theta(n^{\log_b a}) \\ \implies & \text{Case 2} \end{aligned}$$

3.

$$\begin{aligned} & f(n) = n^4 \\ \implies & f(n) = \Omega(n^3) \\ \implies & f(n) = \Omega(n^{\log_b a + \epsilon}) \quad \text{for } \epsilon = 1 \qquad (1) \end{aligned}$$

and:

$$\begin{aligned} af(n/b) &= 2f(n/8) \\ &= \frac{2}{8^3} n^3 \\ &< cf(n) = cn^3 \quad \text{for } c = \frac{3}{8^3} \qquad (2) \end{aligned}$$

(1) and (2) $\implies$ Case 3.

4.

$$f(n) = \frac{1}{\log n} * n^3$$

$\implies \big(f(n) \neq \Theta(n^3)\big) \quad \wedge \quad \big(\forall \epsilon : f(n) \neq \Omega(n^{3+\epsilon})\big) \quad \wedge \quad \big(\forall \epsilon : f(n) \neq O(n^3 - \epsilon)\big)$

$\implies$ The Master theorem does not apply.

5.

$$f(n) = \log n * n^3$$

$\implies \big(f(n) \neq \Theta(n^3)\big) \quad \wedge \quad \big(\forall \epsilon : f(n) \neq \Omega(n^{3+\epsilon})\big) \quad \wedge \quad \big(\forall \epsilon : f(n) \neq O(n^3 - \epsilon)\big)$

$\implies$ The Master theorem does not apply.

Question 3

Algorithm:

1. We iterate over the elements of the set A, and we build up the partial sums into an array S.

2. For each possible pair of indices i, j where $1 \leq i < j \leq n$, compute the difference $s[j] - s[i]$ and check if it equals k.