



# Design and Analysis of Algorithms

By Aditya Anil. AM.ENU4AIE19006

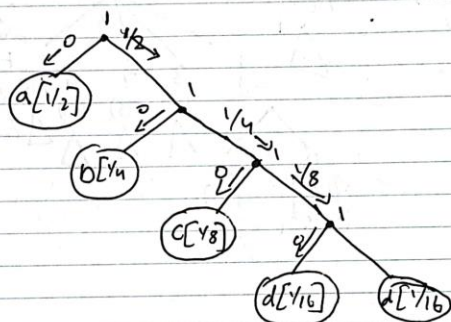
### Question 1

[15mins] Suppose the symbols a, b, c, d, e occur with frequencies  $1/2, 1/4, 1/8, 1/16, 1/16$ , respectively.

- What is the Huffman encoding of the alphabet?
- If this encoding is applied to a file consisting of 1 million chars with the given frequencies, what is the length of the encoded file in bits.

### Answer

(a)  $f_a = 1/2, f_b = 1/4, f_c = 1/8, f_d = 1/16, f_e = 1/16$



From the Huffman encoding tree, the resulting encoding is

CHARACTER	CODEWORD
a	0
b	10
c	110
d	1110
e	1111

$$\begin{aligned}
 \text{(b) Length} &= \left[ \frac{1}{2} \times 1 + \frac{1}{4} \times 2 + \frac{1}{8} \times 3 + \frac{1}{16} \times 4 + \frac{1}{16} \times 4 \right] \times 10^6 \\
 &= \left[ \frac{1}{2} + \frac{1}{2} + \frac{3}{8} + \frac{1}{4} + \frac{1}{4} \right] \times 10^6 \\
 &= \left[ 1 + \frac{1}{2} + \frac{3}{8} \right] \times 10^6 = 1.875 \times 10^6 \\
 &= 1875000
 \end{aligned}$$

### Question 2

[15mins] Consider alphabet {a, b, c} with frequencies  $f_a, f_b$  and  $f_c$ . Given below are the codes for the alphabet.

- Code: [0, 10, 11]
- Code: [0, 1, 00]
- Code: [10, 01, 00]

In each of the above cases, either give an example of frequencies ( $f_a$ ,  $f_b$ ,  $f_c$ ) that would yield the specified code, or explain why the code cannot possibly be obtained (no matter what the frequencies are)

### Answer

$$Q. \text{ Given } = \{a, b, c\}$$

$f_i$  = frequency of  $i$ , where  $i \in$  The given set

(a) code  $\{0, 10, 11\}$   
 $(f_a, f_b, f_c) = \{0.5, 0.25, 0.25\}$

(b) code  $\{0, 1, 00\}$   
 The following code cannot be encoded  
 because code for  $a(0)$  is prefix code  $c(00)$

(c) Code:  $\{10, 01, 00\}$   
 This is not the optimal code as it will result in a shorter encoding. Also it does not correspond to the full binary tree and can not be encoded by Huffman algorithm.

## Question 4

[15mins] Complete the longest increasing subsequence example taught in the class.

## Answer

f	1	2	3	4	5	6	7	8	9	10	11	12	13
$S[i]$	2	5	-1	8	0	4	6	11	7	9	12	10	14
$LISL[i]$	1	2	1	3	2	3	4	5	5	6	7	7	8

$LISL[1] = 1$  (initialisation step)  
 $LISL[2]$   
 $i=2, j=1 \quad S[2] > S[1]$   
 $LISL[2] = 2$   
 $LISL[3]$   
 $i=3, j=1 \quad S[3] \leq S[1]$   
 $i=3, j=2 \quad S[3] \leq S[2]$   
 $LISL[3] = 1$   
 $LISL[4]$   
 $i=4, j=1 \quad S[4] > S[1]$   
 $i=4, j=2 \quad S[4] > S[2]$   
 $i=4, j=3 \quad S[4] > S[3]$   
 $LISL[4] = 3$   
 $LISL[5]$   
 $i=5, j=1 \quad S[5] \leq S[1]$   
 $i=5, j=2 \quad S[5] \leq S[2]$   
 $i=5, j=3 \quad S[5] > S[3]$   
 $i=5, j=4 \quad S[5] \leq S[4]$   
 ~~$LISL[5] = 4$~~   
 $LISL[5] = 1 + 1 = 2$   
 $LISL[6]$   
 $i=6, j=1 \quad S[6] > S[1]$   
 $i=6, j=2 \quad S[6] \leq S[2]$   
 $i=6, j=3 \quad S[6] > S[3]$   
 $i=6, j=4 \quad S[6] \leq S[4]$   
 $i=6, j=5 \quad S[6] > S[5]$   
 $LISL[6] = 2 + 1 = 3 \quad \max\{LISL[3], LISL[6]\} = LISL[6] = 3$

$x = L[SL[7]]$

$SL$

$$i=7, j=1 \quad s[7] > s[1]$$

$$i=7, j=2 \quad s[7] > s[2]$$

$$i=7, j=3 \quad s[7] > s[3]$$

$$i=7, j=4 \quad s[7] \leq s[4]$$

$$i=7, j=5 \quad s[7] > s[5]$$

$$i=7, j=6 \quad s[7] > s[6]$$

$$L[SL[7]] = 1 + \max\{1, 2, 1, 3, 3\} = 4$$

$L[SL[8]]$

$$i=8, j=1 \quad s[8] > s[1]$$

$$i=8, j=2 \quad s[8] > s[2]$$

$$i=8, j=3 \quad s[8] > s[3]$$

$$i=8, j=4 \quad s[8] > s[4]$$

$$i=8, j=5 \quad s[8] > s[5]$$

$$i=8, j=6 \quad s[8] > s[6]$$

$$i=8, j=7 \quad s[8] > s[7]$$

$$L[SL] = 1 + \max\{1, 2, 1, 3, 3, 4\} = 5$$

$L[SL a]$

$$i=9, j=1 \quad s[a] > s[1]$$

$$i=9, j=2 \quad s[a] > s[2]$$

$$i=9, j=3 \quad s[a] > s[3]$$

$$i=9, j=4 \quad s[a] \leq s[4]$$

$$i=9, j=5 \quad s[a] > s[5]$$

$$i=9, j=6 \quad s[a] > s[6]$$

$$i=9, j=7 \quad s[a] > s[7]$$

$$i=9, j=8 \quad s[a] \leq s[8]$$

$$L[LS] = 1 + \max\{1, 2, 1, 2, 3, 4\} = 5$$



$LISL(10)$

$i=10, j=1 \quad s[10] > s[1]$   
 $i=10, j=2 \quad s[10] > s[2]$   
 $i=10, j=3 \quad s[10] > s[3]$   
 $i=10, j=4 \quad s[10] > s[4]$   
 $i=10, j=5 \quad s[10] > s[5]$   
 $i=10, j=6 \quad s[10] > s[6]$   
 $i=10, j=7 \quad s[10] > s[7]$   
 $i=10, j=8 \quad s[10] \leq s[8]$   
 $i=10, j=9 \quad s[10] > s[9]$

$LISL = 1 + \max \{1, 2, 4, 3, 2, 3, 4, 5\} = 6$

$LISL(11)$

$i=11, j=1 \quad s[11] > s[1]$   
 $i=11, j=2 \quad s[11] > s[2]$   
 $i=11, j=3 \quad s[11] > s[3]$   
 $i=11, j=4 \quad s[11] > s[4]$   
 $i=11, j=5 \quad s[11] > s[5]$   
 $i=11, j=6 \quad s[11] > s[6]$   
 $i=11, j=7 \quad s[11] > s[7]$   
 $i=11, j=8 \quad s[11] > s[8]$   
 $i=11, j=9 \quad s[11] > s[9]$   
 $i=11, j=10 \quad s[11] > s[10]$

$LISL(i) = 1 + \max \{1, 2, 1, 3, 2, 3, 4, 5, 6\} = 7$

$LISL(12)$

$i=12, j=1 \quad s[12] > s[1]$   
 $i=12, j=2 \quad s[12] > s[2]$   
 $i=12, j=3 \quad s[12] > s[3]$   
 $i=12, j=4 \quad s[12] > s[4]$   
 $i=12, j=5 \quad s[12] > s[5]$   
 $i=12, j=6 \quad s[12] > s[6]$   
 $i=12, j=7 \quad s[12] > s[7]$   
 $i=12, j=8 \quad s[12] > s[8]$   
 $i=12, j=9 \quad s[12] > s[9]$   
 $i=12, j=10 \quad s[12] > s[10]$   
 $i=12, j=11 \quad s[12] > s[11]$

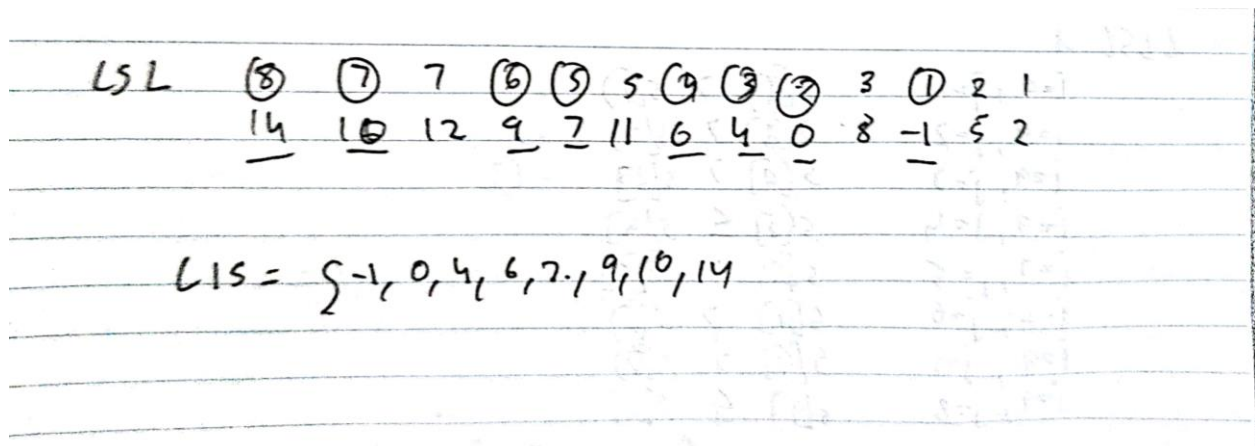
$LISL(12) = 1 + \max \{4, 3, 1, 3, 2, 3, 4, 5, 6\} = 7$

$\rightarrow LISL(13)$

$i=13, j=1 \quad \text{and} \quad s[13] > s[1]$   
 $i=13, j=2 \quad \text{and} \quad s[13] > s[2]$   
 $i=13, j=3 \quad \text{and} \quad s[13] > s[3]$   
 $i=13, j=4 \quad \text{and} \quad s[13] > s[4]$   
 $i=13, j=5 \quad \text{and} \quad s[13] > s[5]$   
 $i=13, j=6 \quad \text{and} \quad s[13] > s[6]$   
 $i=13, j=7 \quad \text{and} \quad s[13] > s[7]$   
 $i=13, j=8 \quad \text{and} \quad s[13] > s[8]$   
 $i=13, j=9 \quad \text{and} \quad s[13] > s[9]$   
 $i=13, j=10 \quad \text{and} \quad s[13] > s[10]$   
 $i=13, j=11 \quad \text{and} \quad s[13] > s[11]$   
 $i=13, j=12 \quad \text{and} \quad s[13] > s[12]$

$LISL(13) = 1 + \max \{1, 2, 1, 3, 2, 3, 4, 5, 6, 7, 7\} = 8$

$LISL(13) = 8$



### Question 5

[30 mins] Submit the dynamic programming solution for the LIS length problem at <https://leetcode.com/problems/longest-increasing-subsequence/>. Submit the code and the proof of your submission.

### Answer

```
class Solution:
    def lengthOfLIS(self, nums: List[int]) -> int:
        arr=nums
        n = len(nums)
        lis = [1]*n
        for i in range (1, n):
            for j in range(0, i):
                if nums [i] > nums [j] and lis[i]< lis[j] + 1 :
                    lis[i] = lis[j]+1
        maximum = 0

        # Pick maximum of all LIS values
        for i in range(n):
            maximum = max(maximum, lis[i])

        return maximum
```

## Longest Increasing Subsequence

### Submission Detail

54 / 54 test cases passed.

Runtime: 2764 ms

Memory Usage: 14.6 MB

Submit

- adiaux
- My List
- My Playground
- Notebook
- Submissions
- Sessions
- Progress
- Points
- Subscription
- Orders
- Sign out

### Accepted Solutions Runtime Distribution

