Homework 4

Arjun Koshal

November 26, 2022

Problem 1

Consider an array containing the following 40 integers: (10 points)

5 2 4 4 0 1 6 7 3 1 1 0 5 1 5 4 4 5 7 0 6 1 0 7 5 2 7 6 5 3 7 0 5 5 7 1 1 2 6 5

Apply counting sort to sort this array and show all the steps of your work.

Solution:

Step 1: Traverse array to find maximum value: 7.
Step 2: Allocate array of size max + 1: 8.

Value	0	1	2	3	4	5	6	7
Counters	0	0	0	0	0	0	0	0

Step 3: Traverse the array again and count number of occurrences for each value.

Value	0	1	2	3	4	5	6	7
Counters	5	7	3	2	4	9	4	6

Step 4: Traverse the array of counters and display the output.

Output: 0000011111112223344445555555556666777777

Value	0	1	2	3	4	5	6	7
Counters	0	0	0	0	0	0	0	0

Problem 2

Consider an array containing the following hexadecimal numbers: (10 points)

4EC1EEA9
520B6E78
1E90D74E
52DB6E42
5F05EF13
74284442
794E8117
55526E42

Consider you are using a version of RadixSort that sorts on one byte at a time (two hexadecimal digits at a time). Under the stated circumstances, show all the steps when you apply radix sort on this array.

Solution:

Convert from base 16 to base 10:

Key:
A = 10
B = 11
C = 12
D = 13
E = 14
F = 15

1st Sort:
Step 1: Convert 6th and 7th Digits

1. $A9 = (10 \cdot 16^1) + (9 \cdot 16^0) = 169$
2. $78 = (7 \cdot 16^1) + (8 \cdot 16^0) = 120$
3. $4E = (4 \cdot 16^1) + (14 \cdot 16^0) = 78$
4. $42 = (4 \cdot 16^1) + (2 \cdot 16^0) = 66$
5. $13 = (1 \cdot 16^1) + (3 \cdot 16^0) = 19$
6. $42 = (4 \cdot 16^1) + (2 \cdot 16^0) = 66$
7. $17 = (1 \cdot 16^1) + (7 \cdot 16^0) = 23$
8. $42 = (4 \cdot 16^1) + (2 \cdot 16^0) = 66$

Step 2: Sort the 1st Cycle

1. 5F05EF13
2. 794E8117
3. 52DB6E42
4. 74284442
5. 55526E42
6. 1E90D74E
7. 520B6E78
8. 4EC1EEA9

2nd Sort:
Step 1: Convert 4th and 5th Digits

1. EF $= (14 \cdot 16^1) + (15 \cdot 16^0) = 239$
2. 81 $= (8 \cdot 16^1) + (1 \cdot 16^0) = 129$
3. 6E $= (6 \cdot 16^1) + (14 \cdot 16^0) = 110$
4. 44 $= (4 \cdot 16^1) + (4 \cdot 16^0) = 68$
5. 6E $= (6 \cdot 16^1) + (14 \cdot 16^0) = 110$
6. D7 $= (13 \cdot 16^1) + (7 \cdot 16^0) = 215$
7. 6E$= (6 \cdot 16^1) + (14 \cdot 16^0) = 110$
8. EE $= (14 \cdot 16^1) + (14 \cdot 16^0) = 238$

Step 2: Sort the 2nd Cycle

1. 74284442
2. 52DB6E42
3. 55526E42
4. 520B6E78
5. 794E8117
6. 1E90D74E
7. 4EC1EEA9
8. 5F05EF13

3rd Sort:
Step 1: Convert 2nd and 3rd Digits

1. $28 = (2 \cdot 16^1) + (8 \cdot 16^0) = 40$
2. $DB = (13 \cdot 16^1) + (11 \cdot 16^0) = 219$
3. $52 = (5 \cdot 16^1) + (2 \cdot 16^0) = 82$
4. $0B = (0 \cdot 16^1) + (12 \cdot 16^0) = 11$
5. $4E = (4 \cdot 16^1) + (14 \cdot 16^0) = 78$
6. $90 = (9 \cdot 16^1) + (0 \cdot 16^0) = 144$
7. C1$= (12 \cdot 16^1) + (1 \cdot 16^0) = 193$
8. $05 = (0 \cdot 16^1) + (5 \cdot 16^0) = 5$

Step 2: Sort the 3rd Cycle

1. 5F05EF13
2. 520B6E78
3. 74284442
4. 794E8117
5. 55526E42
6. 1E90D74E
7. 4EC1EEA9
8. 52DB6E42

4th Sort:
Step 1: Convert 0th and 1st Digits

1. $5F = (5 \cdot 16^1) + (15 \cdot 16^0) = 95$
2. $52 = (5 \cdot 16^1) + (2 \cdot 16^0) = 82$
3. $74 = (7 \cdot 16^1) + (4 \cdot 16^0) = 116$
4. $79 = (7 \cdot 16^1) + (9 \cdot 16^0) = 121$
5. $55 = (5 \cdot 16^1) + (5 \cdot 16^0) = 85$
6. $1E = (1 \cdot 16^1) + (15 \cdot 16^0) = 30$
7. 4E$= (4 \cdot 16^1) + (15 \cdot 16^0) = 78$

8. $52 = (5 \cdot 16^1) + (2 \cdot 16^0) = 82$

Step 2: Sort the 4th Cycle

1. 1E90D74E
2. 4EC1EEA9
3. 520B6E78
4. 52DB6E42
5. 55526E42
6. 5F05EF13
7. 74284442
8. 794E8117

Therefore, the sorted array is:

1. 1E90D74E
2. 4EC1EEA9
3. 520B6E78
4. 52DB6E42
5. 55526E42
6. 5F05EF13
7. 74284442
8. 794E8117

Problem 3

(a) Construct a Huffman tree (variable-length encoding) for the following: (3 points)

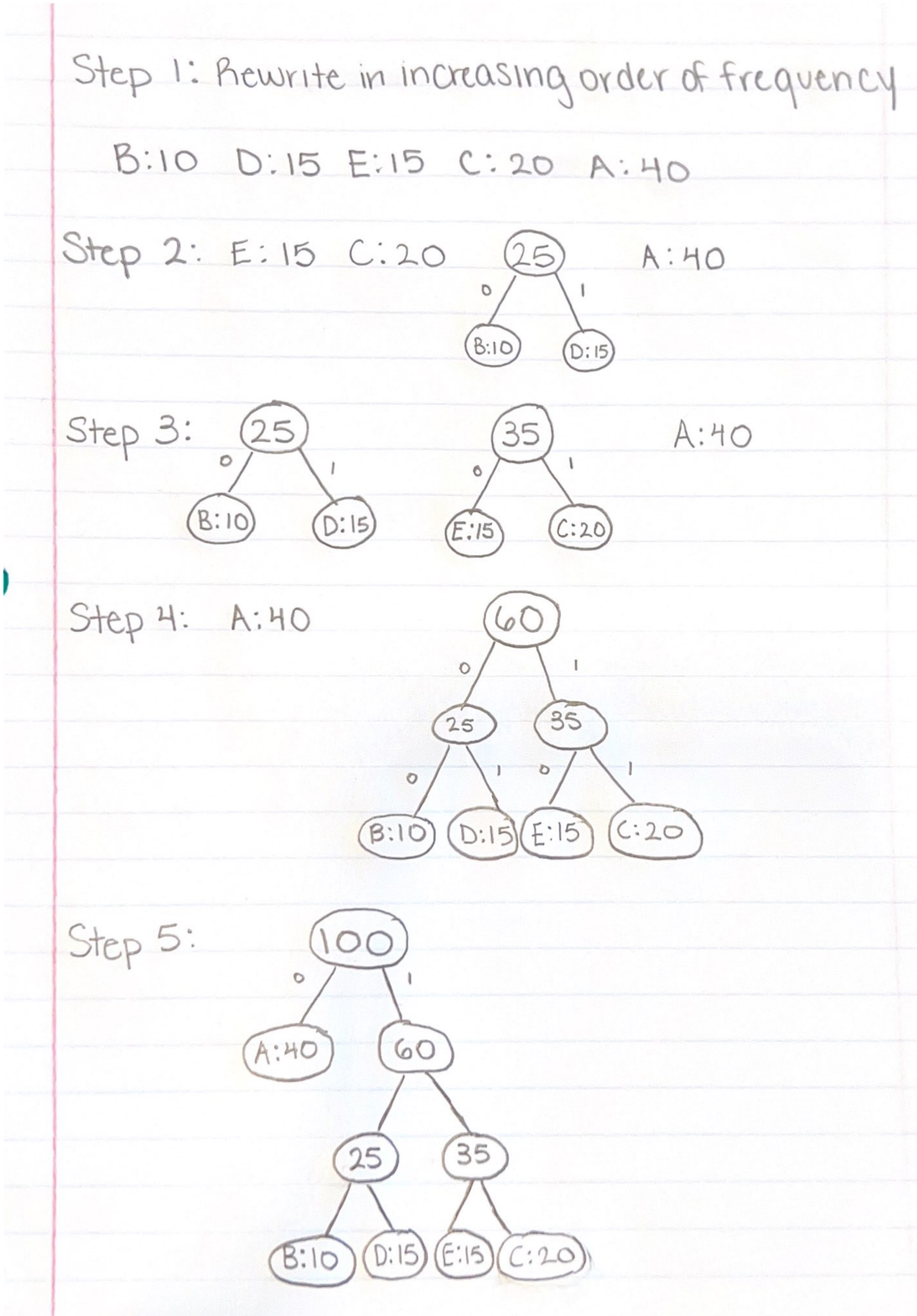
Symbol	A	B	C	D	E
Frequency	40	10	20	15	15

(b) Encode ABACABAD using the tree you generated for (a). (1 point)

(c) Decode 100010111001010 using the tree you generated for (a). (1 point)

(d) What compression gain (percent of improvement) do we get by using Huffman encoding (variable-length encoding) instead of a fixed-length encoding scheme. Draw the tree for the fixed-length encoding. (5 points)

Solution:


Step 1: Rewrite in increasing order of frequency
B:10 D:15 E:15 C:20 A:40
Step 2: E:15 C:20 25 A:40
0 1
B:10 D:15
Step 3: 25 35 A:40
0 1 0 1
B:10 D:15 E:15 C:20
Step 4: A:40 60
0 1
25 35
0 1 0 1
B:10 D:15 E:15 C:20
Step 5: 100
0 1
A:40 60
25 35
B:10 D:15 E:15 C:20

(a)

(b)

Key:

A = 0

B = 100

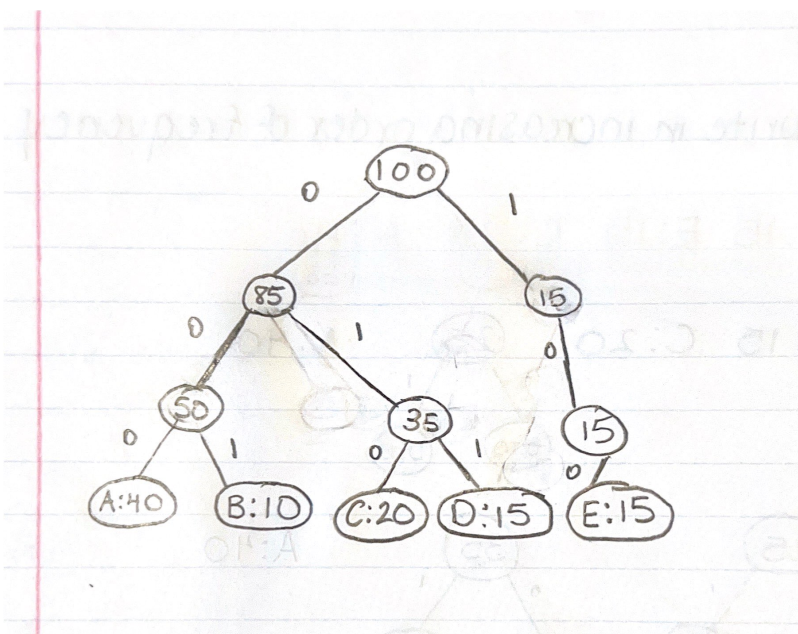
C = 111

D = 101

E = 110

$\boxed{\text{ABACABAD: 0100011101000101}}$

(c) $\boxed{\text{100010111001010: BADEADA}}$



(d)

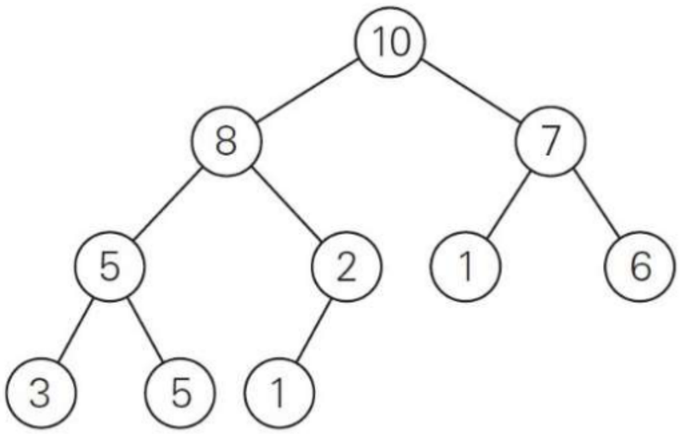
With Fixed Length, we need to use 3 bits (edges) for each character:
$100000 \cdot 3 = 300000$ bits.

With Variable Length, we need to use a different number of bits for each character:
$(40000 \cdot 1) + (10000 \cdot 3) + (20000 \cdot 3) + (15000 \cdot 3) + (15000 \cdot 3) = 220000$ bits.

Therefore the savings with respect to fixed length: $\frac{300000-220000}{300000} = \boxed{26.67\%}$.

Problem 4

Consider the following binary tree.



(a) Traverse the tree preorder. (2 points)

(b) Traverse the tree inorder. (2 points)

(c) Traverse the tree postorder. (2 points)

(d) How many internal nodes are there? (1 point)

(e) What is the maximum width of the tree? (1 point)

(f) What is the height of the tree? (1 point)

(g) What is the diameter of the tree? (1 point)

Solution:

(a) 10 8 5 3 5 2 1 7 1 6

(b) 3 5 5 8 1 2 10 1 7 6

(c) 3 5 5 1 2 8 1 6 7 10

(d) 5 internal nodes (10, 8, 5, 2 and 7 are not leaves)

(e) 4 is the maximum width since the maximum number of nodes with the same depth is at a depth 2 which contains the nodes 5, 2, 1 and 6

(f) 3 is the height of the tree since the number of nodes between the root (10) and its furthest leaf (3, 5, 1) is 3 edges

(g) 5 is the diameter since the length or number of edges between nodes 3 and 1 for the right subtree is 5 edges

Problem 5

Draw the 2-3 tree after inserting each of the following keys. Redraw the whole tree for each part.

(a) 50 (1 point)

(b) 76 (1 point)

(c) 23 (2 points)

(d) 21 (2 points)

(e) 20 (2 points)

(f) 19 (2 points)

Solution: