National University of Computer and Emerging Sciences

FAST School of Computing **Fall-2022** **Islamabad Campus**

CS-2009: Design and Analysis of Algorithms (SE P,Q,R,S)

Saturday, 24th December, 2022

Course Instructor

Noor ul Ain, Bilal Khalid Dar

Serial No:

Final Exam

Total Time: 3 Hours

Total Marks: 120

Signature of Invigilator

Student Name Roll No. Section Signature

DO NOT OPEN THE QUESTION BOOK OR START UNTIL INSTRUCTED.

Instructions:

1. Attempt on question paper. Attempt all of them. Read the question carefully, understand the question, and then attempt it. In case of any ambiguity write down your assumption and solve the question.
2. No additional sheet will be provided for rough work. Use the back of the last page for rough work.
3. If you need more space write on the back side of the paper and clearly mark question and part number etc.
4. After asked to commence the exam, please verify that you have fourteen (14) different printed pages including this title page. There are a total of 10(Ten) questions.
5. Calculator sharing is strictly prohibited.
6. Use permanent ink pens only. Any part done using soft pencil will not be marked and cannot be claimed for rechecking.

	Q-1	**Q-2**	**Q-3**	**Q-4**	**Q-5**	**Q-6**	**Q-7**	**Q-8**	**Q-9**	**Q-10**	**Total**
Marks Obtained											
Total Marks	**10**	**12**	**11**	**14**	**15**	**10**	**10**	**10**	**13**	**15**	**120**

Question 1 [10 Marks]

a) **Provide a worst-case asymptotic time complexity of the following algorithms by using a suitable asymptotic notation considering a nearest function. Assume that there are no errors/ bugs in the algorithms. Show the meaningful working behind your answer.**

Code	Time Complexity

int sum=0;
for (int i = 0; i < n ; i+=2)
{ if (i % 10 == 0)
  {
    for (int j = 0; j < i; j++)
      sum++
  }
 }	

for (int i = 0; i < n * n; ++i)
{
  for (int k = 0; k < i; ++k)
      cout<<k;
  for (int j = n; j > 0; j--)
      cout<<j;
}	

i = n;
 while(i > 1)
 {
  j = i;   //this does not start at 0
    while (j < n)
     { k = 0;
          while (k < n)
             k = k + 2;
          j = j * 2;
     }
       i = i / 2;
  }	

Question 2 [12 Marks]

a) Sort all the functions below in increasing order of asymptotic (big-O) growth. If some have the same asymptotic growth, then be sure to indicate that. As usual, lg means base 2. [2 marks]

10^n , $n^{1/3}$, 2^{2n}, n^{20}, lgn, $n!$, 2^{2^n}, $\sqrt{n}$

b) Show that $3n^2 + n + 1$ is $\Theta(n^2)$ by directly finding the constants k, C1, and C2 [5 points]

c) Derive the recurrence relation that describes processing time T(n) of the recursive method given in the table below: [5 marks]

```
function finalExam ( n )
        if (n > 1)
             print 'A'
            finalExam( n /3 )
                  for i = 1 to n
                       print 'B'
                 end for
            finalExam ( n /3 )
```

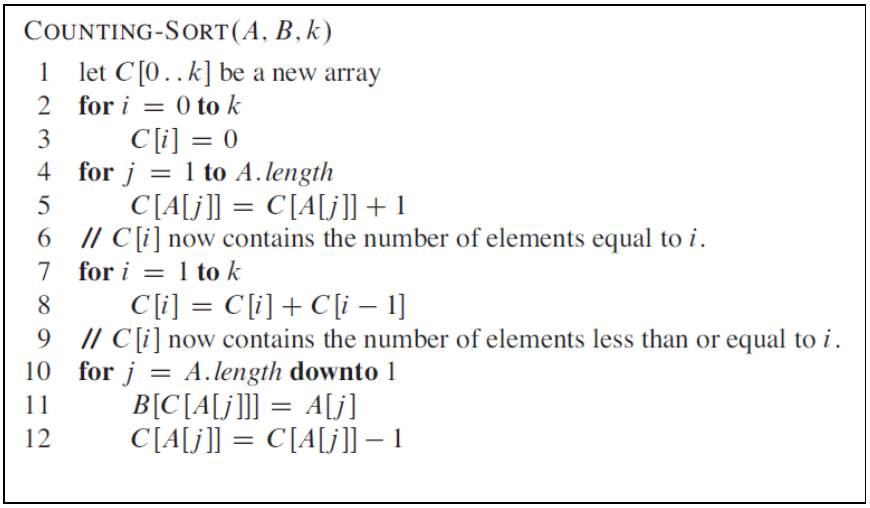
Provide Recurrence Relation of the above pseudocode including base case: [2 marks]

What is the runtime of the above function? Express your answer using the big-O notation [3 marks]

Question 3 [11 Marks]

Counting Sort is known as a stable sorting algorithm. **table sort** is described as a sorting algorithm that *maintains the position of two equals elements relative to one another.* That is, a sorting algorithm is stable if whenever there are two records R and S with the same key and with R appearing before S in the original list, R will appear before S in the sorted list.

Provide an array of at least 7 elements to demonstrate that the Counting Sort is a stable sort. Also provide complete dry run of counting sort on your selected array. [6 marks]



```
COUNTING-SORT($A, B, k$)
let $C[0..k]$ be a new array
for $i = 0$ to $k$
    $C[i] = 0$
for $j = 1$ to $A.length$
    $C[A[j]] = C[A[j]] + 1$
// $C[i]$ now contains the number of elements equal to $i$.
for $i = 1$ to $k$
    $C[i] = C[i] + C[i-1]$
// $C[i]$ now contains the number of elements less than or equal to $i$.
for $j = A.length$ downto 1
    $B[C[A[j]]] = A[j]$
    $C[A[j]] = C[A[j]] - 1$
```

b) What is the best scenario to use Counting sort? Is Quick Sort a stable sort? Why/Why not? Explain with the help of an example. [3 marks]

c) For sorting an input list of size *32* by QuickSort recursive algorithm, how many QuickSort calls will be made? Presume the Pivot choice policy is perfect, and QuickPartition always divides the list in half. Explain in one or two lines. [2 marks]

Question 4 [14 Marks]

A d-ary heap is like a binary heap, but (with one possible exception) non-leaf nodes have d children instead of 2 children.

1. Draw d-ary min heap with minimum 2 levels by specifying any value of 'd' other than 2? [5 marks]

2. How would you represent a d-ary heap in an array? [4 marks]

3. What will be the formulas for finding the parent and children for a given index? [4 marks]

4. What is the height of a d-ary heap of n elements in terms of n and d? [1 marks]

Question 5 [15 Marks]

The following information is based on a piece of text using a set of five different symbols. The frequencies of the symbols in the text are given below:

Symbol	Frequency
A	6
B	18
C	25
D	11
E	03

a) What is the minimum number of bits required to store the text using a fixed-length coding scheme? Justify your answer. [3 Marks]

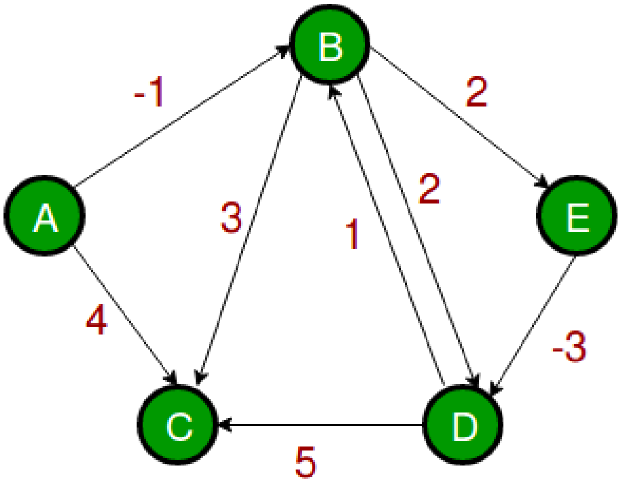
b) What is the minimum number of bits required to store the text using a variable-length coding scheme? You are required to use the Huffman's algorithm learnt in the class. Justify your answer by showing all steps. [10 Marks]

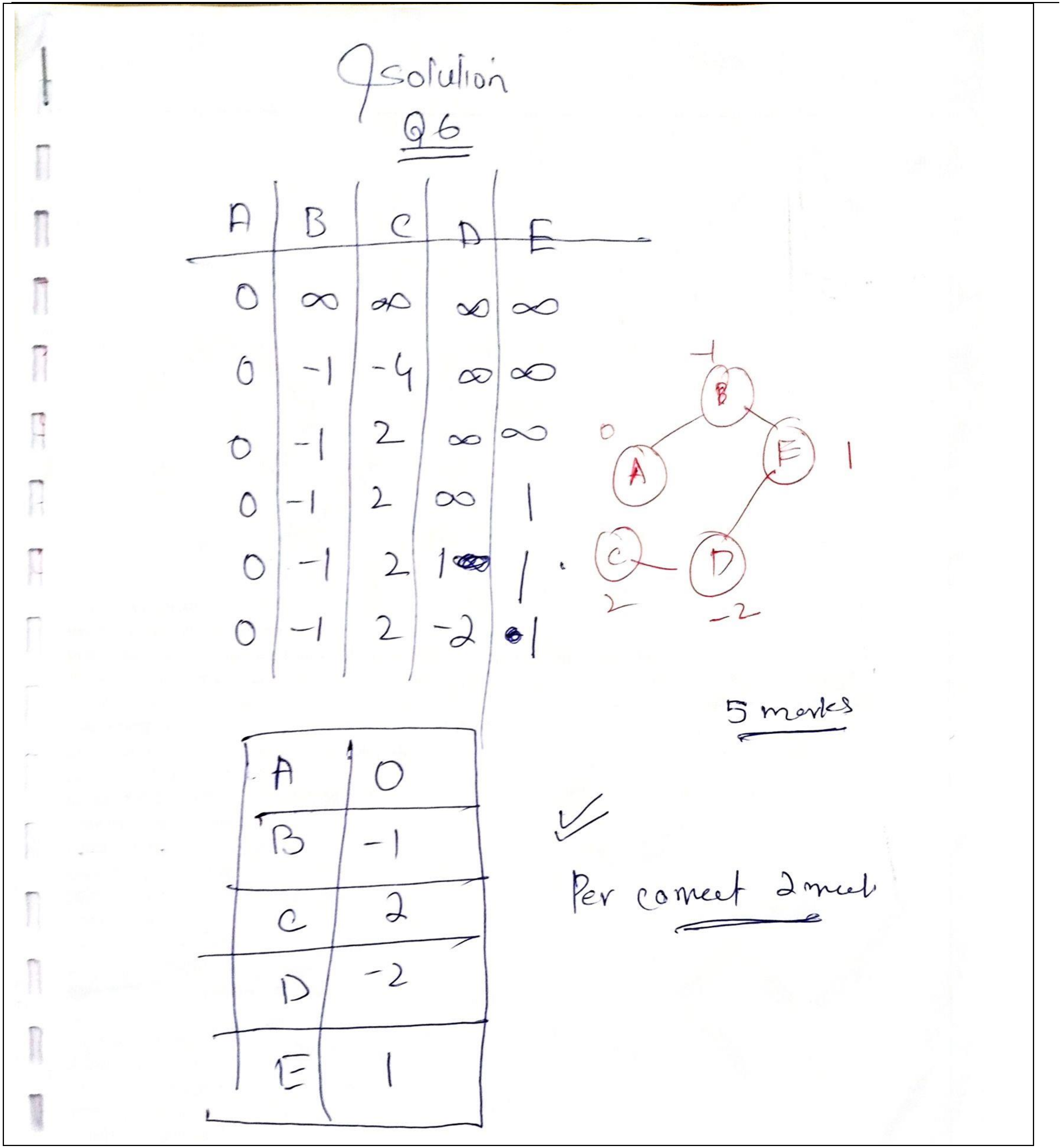
c) Show the final Huffman's tree. **[2 Marks]**

Fill this table	
Symbol	**Variable length code**
A	
B	
C	
D	
E	

Question 6 [10 Marks]

Finding the shortest distance to all vertices from the source using the Bellman-Ford algorithm for the following graph where starting vertex (source) is A





Solution

Q6

A	B	C	D	E
0	∞	∞	∞	∞
0	−1	−4	∞	∞
0	−1	2	∞	∞
0	−1	2	∞	1
0	−1	2	1 ~~∞~~	1
0	−1	2	−2	1

-1
B
0
A
E
1
C
D
2
-2

5 marks

A	0
B	−1
C	2
D	−2
E	1

✓

Per correct 2 mark

Question 7 [10 Marks]

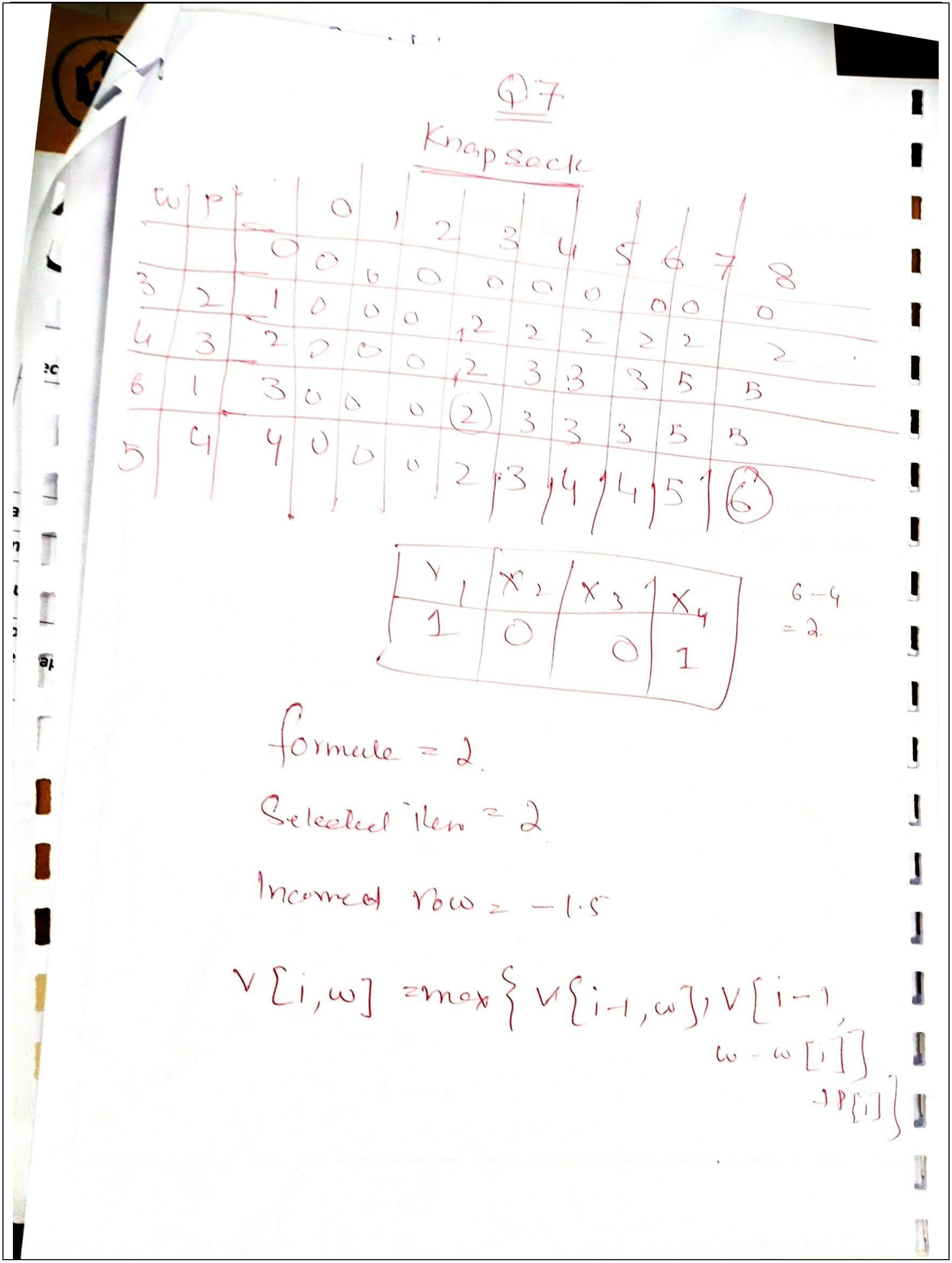
Knapsack 0-1 problem

For the given 4 items below, find most valuable subset of the items that fit into the 0-1knapsack of capacity 8kg using dynamic programming technique?

item	weight(kg)	value
1	3	$ 2
2	4	$ 3
3	6	$ 1
4	5	$ 4

Recursive Formula: **[2 marks]**

Fill the table using the recursive definition for solving the 0-1 Knapsack problem; [6 marks]



Q7

Knapsack

w	p		0	1	2	3	4	5	6	7	8
		0	0	0	0	0	0	0	0	0	0
3	2	1	0	0	0	2	2	2	2	2	2
4	3	2	0	0	0	2	3	3	3	5	5
6	1	3	0	0	0	$\boxed{2}$	3	3	3	5	5
5	4	4	0	0	0	2	3	4	4	5	$\boxed{6}$

x_1	x_2	x_3	x_4
1	0	0	1

6 − 4
= 2.

formule = 2.

Selected item = 2

Incorrect row = −1.5

$v[i,w] = \max\{v\{i-1,w\}, v[i-1, w-w[i]] + p[i]\}$

Question 8 [10 Marks]

Use dynamic programming technique for rod cutting problem to find the maximum profit for selling the rod of length 8. Following table shows piece length and price of each piece of a rod

Piece Length	1	2	3	4	5	6	7	8
Profit	1	3	4	5	7	9	10	11

Show formula, table and the results

Recursive Formula:	**[2 marks]**
Table:	**[6 marks]**

Q8

PL	1	2	3	4	5	6	7	8
P	1	3	4	5	7	9	10	11

m = 8

	w\	0	1	2	3	4	5	6	7	8
	0	0	0	0	0	0	0	0	0	0
	1	0	1	2	3	4	5	6	7	8
3	2	0	1	3	4	6	7	9	10	12
4	3	0	1	3	4	6	7	9	10	12
5	4	0	1	3	4	6	7	9	10	12
7	5	0	1	3	4	6	7	9	10	12
9	6	0	1	3	4	6	7	9	10	12
10	7	0	1	3	4	6	7	9	10	12
11	8	0	1	3	4	6	7	9	10	12

0 7 5

$P[i-w_i] + P_i$

max (Profit with new piece   Profit without new piece) → $P[i-1]$

→ 2, 2, 2, 2

6, 2 is considered

Final Results (Maximum profit and rod cutting values in pieces) **[2 marks]**

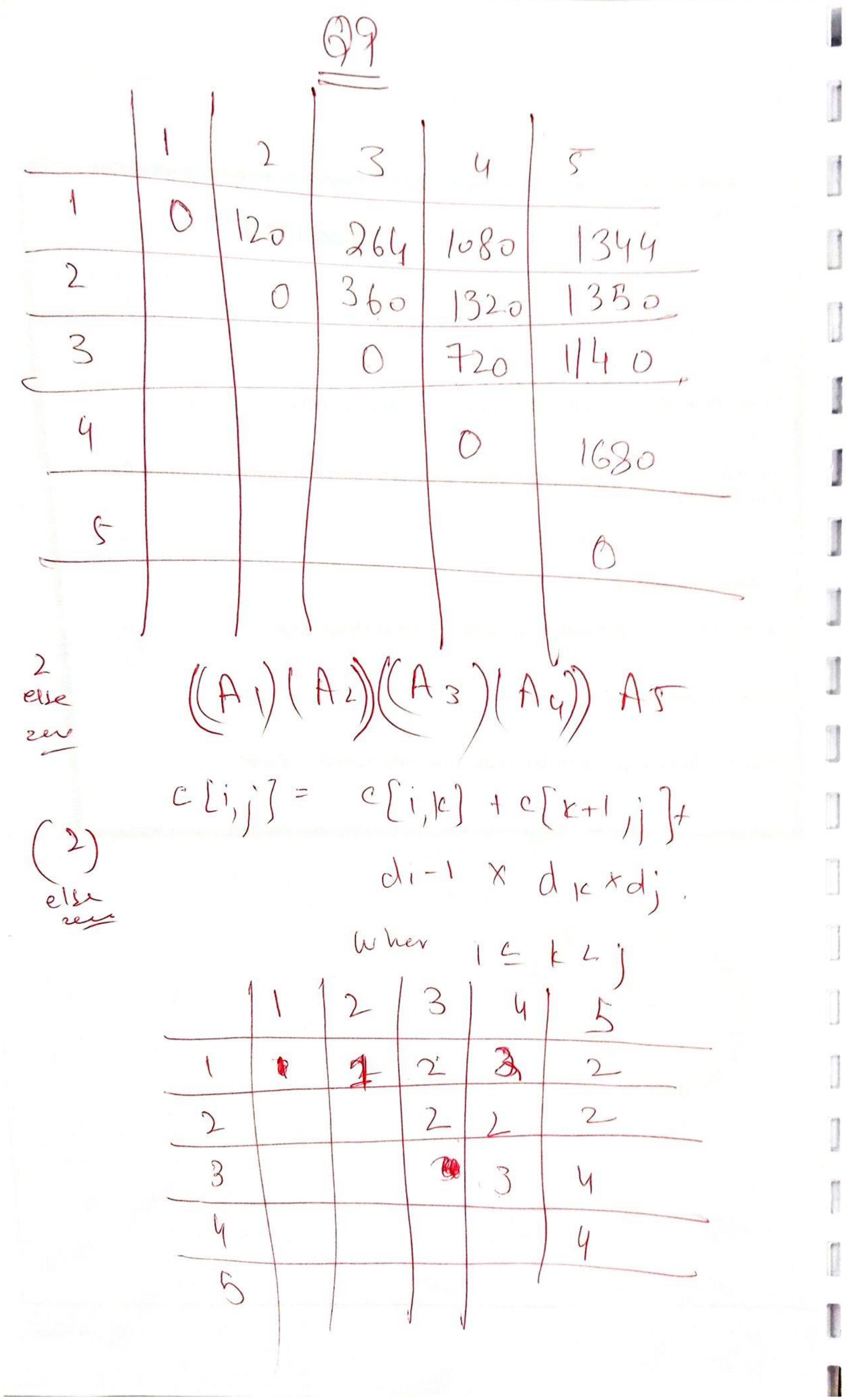
Question 9 [13 Marks]

We have 5 matrixes of following which should be multiplied as A * B * C* D* E. Matrix sizes are

A = 4 x 10,
B = 10 x 3,
C = 3 x 12,
D = 12 x 20,
E = 20 x 7

If multiplication is possible, show the minimum number of multiplications required using dynamic programming. Show the formula, table and the results. Also show the order of the multiplication with the help of parentheses

Recursive Formula:	**[2 marks]**
Table:	**[9 marks]**



Q9

	1	2	3	4	5
1	0	120	264	1080	1344
2		0	360	1320	1350
3			0	720	1140
4				0	1680
5					0

2 else zero

$((A_1)(A_2))((A_3)(A_4))\ A_5$

$c[i,j] = c[i,k] + c[k+1,j] + d_{i-1} \times d_k \times d_j$.

(2) else zero

wher $1 \leq k < j$

	1	2	3	4	5
1		1	2	3	2
2			2	2	2
3				3	4
4					4
5					

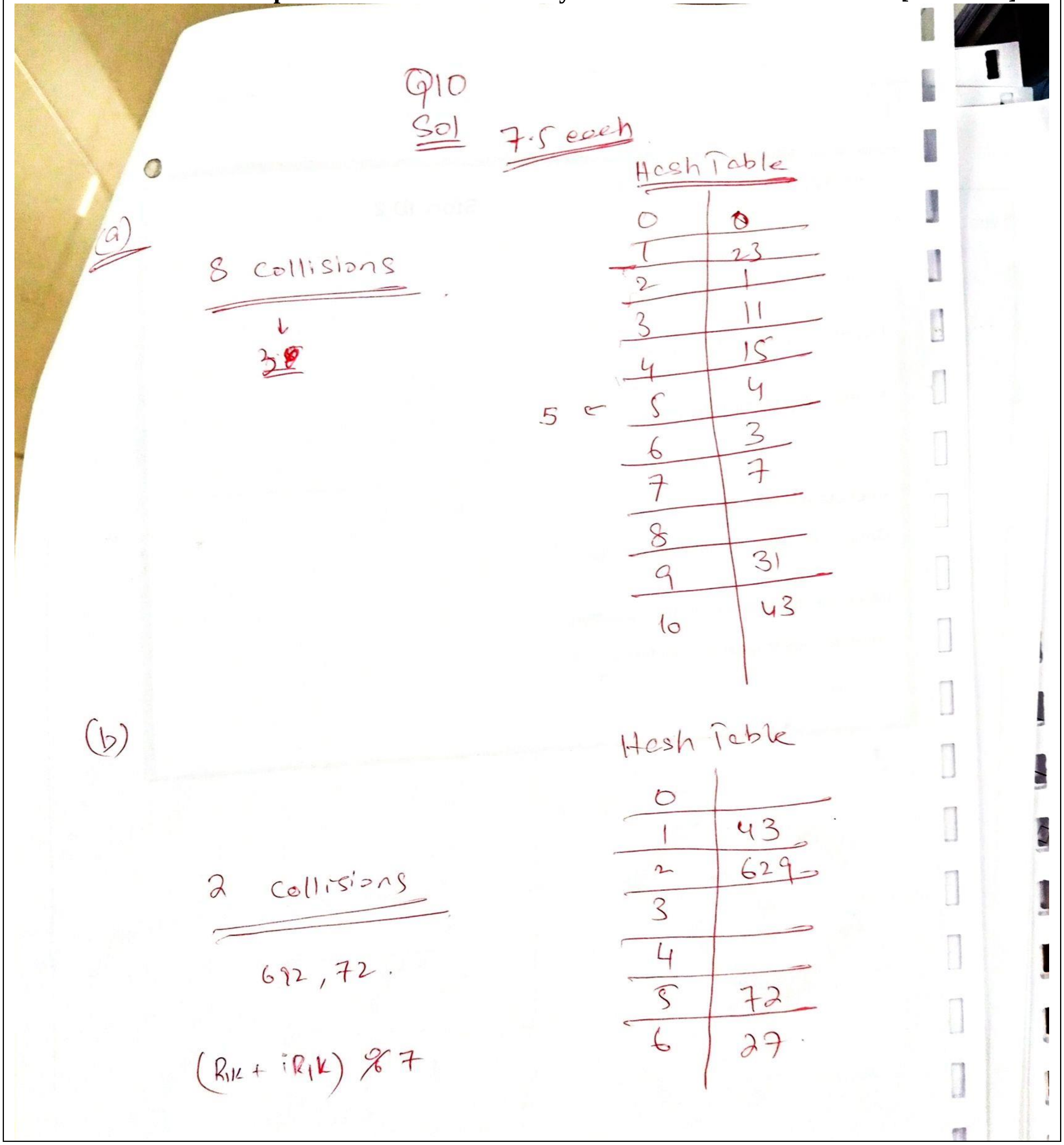
Final Parenthesization: **[2 marks]**

Question 10 [15 Marks]

Consider a hash table consisting of M = 11 slots and suppose nonnegative integer key values are hashed into the table using the hash function key mod 11. Suppose that collisions are resolved by using linear probing. The integer key values listed below are to be inserted, in the order given

Keys	43	23	1	0	15	31	4	7	11	3

Create a hash table and place all the values. Identify the total number of collisions. [08 marks].



Q10

Sol

7.5 each

(a)

8 collisions

↓

3.~~0~~

Hash Table

0	0
1	23
2	1
3	11
4	15
5	4
6	3
7	7
8	
9	31
10	43

5 ←

(b)

Hash Table

0	
1	43
2	629
3	
4	
5	72
6	27

2 collisions

692, 72.

$(R_{1k} + iR_{1k}) \% 7$

a) Insert the keys 27, 43, 692, 72 into the Hash Table of size 7, where the first hash-function is h1(k) = k mod 7 and second hash-function is h2(k) = 1 + (k mod 5). Use double hashing for probing [07 marks]