Assignment #3

Date: M T W T F S S

Design and Analysis of Algorithms.

S. Muhassan Ali
K20-1052
BSE-5B
Group members:
Ehtesham 1655
Amna 1695

Q.1 A = CABA

B = BCBDAB

		B	C	B	D	A	B
	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
C 1	0	0	1	1	1	1	1
A 2	0	0	1	1	1	2	2
B 3	0	1	1	2	2	2	3
A 4	0	1	1	2	2	3	3

Crossing diagonal elements = CAB.

HASSAN
EHTESHAM

		E	H	T	E	S	H	A	M
	0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0	0
H 1	0	0	1	1	1	1	1	1	1
A 2	0	0	1	1	1	1	1	2	2
S 3	0	0	1	1	1	2	2	2	2
S 4	0	0	1	1	1	2	2	2	2
A 5	0	0	1	1	1	2	2	3	3
N 6	0	0	1	1	1	2	2	3	3

if not match take prev and digonal else if match take left diagonal and add 1.

= HSA

20K-1052

Date:

M	T	W	T	F	S	S

b) Shortest Common Supersequence.

		E	H	I	E	S	H	A	M
	0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0	0
H 1	0	0	1	1	1	1	1	1	1
A 2	0	0	1	1	1	1	1	2	2
S 3	0	0	1	1	1	2	2	2	2
S 4	0	0	1	1	1	2	2	2	3
A 5	0	0	1	1	1	2	2	3	3
N 6	0	0	1	1	1	2	2	3	3

nmashsaethe

enteahssamn

Logic is to eliminate the element from x and y as we move diagonally in LCS. Then reversing the whole string.

Page No. _____

Signature _____

20K-1052

Longest Common Subsequence

Date: M T W T F S S

c) Group member 1: 1655
Group member 2: 1695

5, 10, 2, 5, 20

```
while(i > j){
if(arr[i]) > arr[j])
{
Lis[i] = max(Lis[i],
  Ljs[j]+1);
} j++
else { i++; }
}
```

j = 0 ; if i = j

	j		i			
arr =	5	10	2	5	20	
Lis =	1	1	1	1	1	

1

	j →		i			
arr =	5	10	2	5	20	
Lis =	1	1	1	1	1	

2-

	j →		i			
arr =	5	10	2	5	20	
Lis =	1	2	1	1	1	

max(Lis[i], Lis[j+1])
max(1, 1+1)
max(1, 2)

3-

	j →			i		
arr =	5	10	2	5	20	
Lis =	1	2	1	2	1	

4-

	j →				i	
arr =	5	10	2	5	20	
Lis =	1	2	1	2	3	

1 2 3

Lis would be now => (5, 10, 20)
=> (2, 5, 20)

Page No. _____ RC Signature _____

20K-1052

Levenshtein-distance (edit-distance)

Date: M T W T F S S

d) str1 = "PLASMA"
str2 = "ALTRUISM"

	""	A	L	T	R	U	I	S	M
""	0	1	2	3	4	5	6	7	8
P	1	1	2	3	4	5	6	7	8
L	2	2	1	2	3	4	5	6	7
A	3	2	2	2	3	4	5	6	7
S	4	3	3	3	3	4	5	5	6
M	5	4	4	4	4	4	5	6	5
A	6	5	5	5	5	5	5	6	6

Total 6 operations need to perform that could be either update, add, delete for converting str1 to str2.

Logic is to choose previous, upper diagonal and upper no if the characters do not match. Choose the minimum of these and add 1.

If the element matches take upper diagonal only.

$C[i,j] = \min\{C[i,k] + C[k+1,j] + P_{i-1} \times P_k \times P_j\}$ $i \le k < j$

Matrix Chain.

Date: M T W T F S S 2020 052

e) $A_1 \times A_2 \times A_3 \times A_4 \times A_5$

$2 \times 25 \times 25 \times 3 \times 3 \times 16 \times 16 \times 1 \times 1 \times 6$

P_0 P_1 P_2 P_3 P_4 P_5

Hassan

length = 6

	1	2	3	4	5
1	0	150	256	173	185
2		0	1200	123	273
3			0	48	66
4				0	96
5					0

$C[1,2] = \min^{k=1}\{C[1,1] + C[2,2] + P_0P_1P_2$

$0 + 0 + 2 \times 25 \times 3 = 150$

$C[2,3] = \min^{k=2}\{C[2,2] + [3,3] + d_1d_2d_3$

$0 + 0 + 25 \times 3 \times 16 = 1200$

$C[3,4] = \min^{k=3}\{C[3,3] + [4+4] + d_2d_3d_4$

$0 + 0 + 3 \times 16 \times 1 = 48$

$C[4,5] = 96$

$C[1,3] = \min^{k=1}\{C[1,1] + C[2,3] + d_0d_1d_3$

$0 + 1200 + 2 \times 25 \times 16 = 800$

$\checkmark k=2\{C[1,2] + C[3,3] + d_0d_2d_3$

$150 + 0 + 2 \times 3 \times 16 = 96$

Page No. _____ Signature _____

20K-1052.

Date:

M	T	W	T	F	S	S

$$C[2,4] = \min \begin{cases} k=2 & C[2,2] + C[3,4] + d_1 d_2 d_4 \\ & 0 + 48 + 25 \times 3 \times 1 = 123 \\ k=3 & C[2,3] + C[4,4] + d_1 d_3 d_4 \end{cases}$$

$$C[2,5] = \min \begin{cases} k=2 & C[2,2] + C[3,5] + d_1 d_2 d_5 \\ & 0 + 66 + 25 \times 3 \times 6 = 516 \\ k=3 & C[2,3] + C[4,5] + d_1 d_3 d_5 \\ & 1200 + 96 + 25 \times 16 \times 6 = 3696 \\ k=4 & C[2,4] + C[5,5] + d_1 d_4 d_5 \\ & 123 + 0 + 25 \times 1 \times 6 = 273 \end{cases}$$

min

20k-1052.

Date: M T W T F S S

0-1 Knap Sack Problem

f) W = [1, 2, 3, 4, 5] C = 5
V = [3, 5, 4, 8, 10]

bounded.

capacity →

V_i	W_i	0	0	1	2	3	4	5
		0	0	0	0	0	0	0
3	1	1	0	3	3	3	3	3
5	2	2	0	3	5	8	8	8
4	3	3	0	3	5	8	8	9
8	4	4	0	3	5	8	8	11
10	5	5	0	3	5	8	8	11

When considering 2nd object or more consider previous ones too.

$$V[i, w] = \max \{ V[i-1, w], V[i-1, w-w_i] + V[i] \}$$

Partition Problem

20K-1052.

Date: M T W T F S S

g) Hassan Amna (special case of Knapsack)

$S = \{8, 1, 19, 1, 13, 14\}$

Sum = 56/2 = 28

- Find elements that will add up to sum 28, followed by others

```
if (nums[i-1] > j){
    T[i][j] = T[i-1][j];
}
else {
T[i][j] = T[i-1][j] || T[i-1][j-nums[i-1]];
}
```

Finding the subsets for equal sum.

$S_1 = \{8, 1, 19\}$ sum = 28

$S_2 = \{1, 13, 14\}$ sum = 28.

20K-1052.

Date:

M	T	W	T	F	S	S

Rod Cutting Problem.

(unbounded Knap Sack)

h)

Length[] = {1, 2, 3, 4, 5, 6, 7, 8}.

Price[] = {1, 5, 8, 9, 10, 16, 18, 20}

Rod length = 8.

len	1	2	3	4	5	6	7	8
$V_{(i)}$ price	1	5	8	9	10	16	18	20
$C_{(i)}$	1	5	8	10	13	17	18	22

$$C(i) = \max\{V_k + C(i-k)\}$$

$$C(1) = \max\{1 + C(0)\} = 1$$

$$C(2) = \max\begin{cases} 1 + C(2-1) = 1+1 = 2 \\ V_2 = 5 \end{cases} \quad \max = 5$$

$$C(3) = \max\begin{cases} 1 + C(3-1) = 1+5 \\ 5 + C(3-2) = 5+1 \\ V_3 = 8 \end{cases} \quad = 8$$

$$C(4) = \max\begin{cases} 1 + C(4-1) = 1+8 \\ 5 + C(4-2) = 5+5 \\ 8 + C(4-3) = 8+1 \\ V_4 = 9 \end{cases} \quad = 10$$

Page No. _____ Signature _____

20K-1052.

Date:

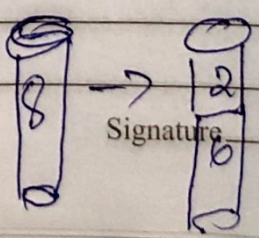
M T W T F S S

$$C(5) = \begin{cases} 1 + C(5-1) & = 1+10 \\ 5 + C(5-2) & = 5+8 \\ 8 + C(5-3) & = 8+5 \\ 9 + C(5-4) & = 9+1 \\ V_5 = 10 & = 10 \end{cases} = 13 \text{ max}$$

$$C(6) = \begin{cases} V_1 + C(5) & = 1+16 \\ V_2 + C(4) & = 5+10 \\ V_3 + C(3) & = 8+8 \\ V_4 + C(2) & = 9+5 \\ V_5 + C(1) & = 10+1 \\ V_6 & = 16 \end{cases} = 17 \text{ max}$$

$$C(7) = \begin{cases} V_1 + C(6) & = 1+17 \\ V_2 + C(5) & = 5+13 \\ V_3 + C(4) & = 8+10 \\ V_4 + C(3) & = 9+8 \\ V_5 + C(2) & = 10+5 \\ V_6 + C(1) & = 16+1 \\ V_7 & = 18 \end{cases} = 18 \text{ max}$$

$$C(8) = \begin{cases} V_1 + C(7) & = 1+18 \\ V_2 + C(6) & = 5+17 \\ V_3 + C(5) & = 8+13 \\ V_4 + C(4) & = 9+10 \\ V_5 + C(3) & = 10+8 \\ V_6 + C(2) & = 16+5 \\ V_7 + C(1) & = 18+1 \\ V_8 & = 20 \end{cases} = 22 \text{ max}$$



Page No.

Signature

20K-1052.

Date:

M	T	W	T	F	S	S

Coin change Making

i) $S = \{1, 5, 6, 8\}$. change = 13.

		0	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	0	1	2	3	4	5	6	7	8	9	10	11	12	13
5	1	0	1	2	3	4	1	2	3	4	5	2	3	4	5
6	2	0	1	2	3	4	1	1	2	3	4	2	2	2	3
8	3	0	1	2	3	4	1	1	2	1	2	2	2	2	2

```
if(coins[i] > j ){
  arr[i][j] = a[i-1][j] }
else {
arr[i][j] = min(a[i-1][j],
    1 + a[i][j-coins[i]]);
```

Min 2 coins required to make desired change of 13.
= {5, 8}.

RC

Signature

20K-1052

Word Break Problem

Date: M T W T F S S

i) S = {i, like, ~~sam, sung~~, mobile, apple, ice, icecream, ~~india~~, ~~go~~, ~~mango~~}

Input = "Ilikeapple" (length = 9)

	0	1	2	3	4	5	6	7	8	9
0	T	F	F	F	T	F	F	F	F	F
1		F	F	F	T	F	F	F	F	T
2			T	F	F	F	F	F	F	F
3				F	F	F	F	F	F	F
4					F	F	F	F	F	F
5						F	F	F	F	T
6							F	F	F	F
7								F	F	F
8									F	F
9										F

length of input = 1

Input(0,0) = i ✓
Input(1,1) = l X
Input(2,2) = i ✓
Input(3,3) = k X
Input(4,4) = e X
Input(5,5) = a X
Input(6,6) = p X
Input(7,7) = p X
Input(8,8) = [illegible] X
Input(9,9) = e X

RC
Signature

20k-1052.

Date:

M	T	W	T	F	S	S

ilikeapple

length of Input = 2.

Input(0,1) = il ✗
Input(1,2) = li ✗
Input(3,4) = ik ✗
Input(5,6) = ke ✓
Input(6,7) = ea ✗
Input(6,7) = ap ✗
Input(7,8) = pp ✗
Input(8,9) = pl ✗
~~Input(10 = le~~ ✗

Input = 3

Input(0,2) = ili ✗
Input(1,3) = lik ✗
Input(2,4) = ike ✗
Input(3,5) = kea ✗
Input(4,6) = eap ✗
Input(5,7) = app ✗
Input(6,8) = ppl ✗
Input(7,9) = ple ✗

length of Input = 4

Input(0,3) = ilik ✗
Input(1,4) = like ✓
Input(2,5) = ikea ✓
Input(3,6) = keap ✗
Input(4,7) = eapp ✗
Input(5,8) = appl ✗
Input(6,9) = pple ✗

Input = 5

Input(0,4) = ilike ✓
Input(1,5) = likea ✗
Input(2,6) = ikeap ✗
Input(3,7) = keapp ✗
Input(4,8) = eappl ✗
Input(5,9) = apple ✓

20k-1052

Date:

M	T	W	T	F	S	S

"ilikeapple"

length = 6

Input(0,5) = "ilikea" ×
Input(1,6) = "likeap" ×
Input(2,7) = "ikeapp" ×
Input(3,8) = "keappl" ×
Input(4,9) = "eapple" ×

length = 7

Input(0,6) = "ilikeap" ×
Input(1,7) = "likeapp" ×
Input(2,8) = "ikeappl" ×
Input(3,9) = "keapple" ✓

length = 8

Input(0,7) = "ilikeapp" ×
Input(1,8) = "likeappl" ×
Input(2,9) = "ikeapple" ×

length = 9

Input(0,8) = "ilikeappl" ×
Input(1,9) = "likeapple" ✓