ASSIGNMENT 3

Date _____

Design and Analysis of Algorithm.

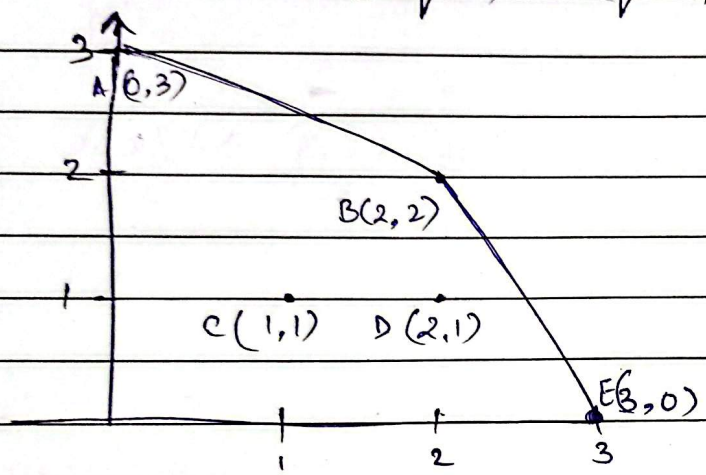
23K-0078

BAI-5A

Q1: A(0,3) B(2,2) C(1,1) D(2,1) E(3,0) F(0,0)

Starting point: F(0,0)

Orientation: $(p,q,r) = (q_x - p_x)(r_y - p_y) - (q_y - p_y)(r_x - p_x)$



Step	Current Point (C)	Candidate (E)	Checking Point (R)	Orientation (P, E, r)	More CCW	New Candidate
1	F(0,0)	A(0,3)	B(2,2)	(0-0)(2-0)-(3-0)(2-0)=-6	CW	A
			C(1,1)	(0-0)(1-0)-(3-0)(1-0)=-3	CW	A
			D(2,1)	(0-0)(1-0)-(3-0)(2-0)=-6	CW	A
			E(3,0)	(0-0)(0-0)-(3-0)(3-0)=-9	CW	A
2	A(0,3)	B(2,2)	C(1,1)	(2-0)(1-3)-(2-3)(1-0)=-3	CW	B
			D(2,1)	(2-0)(1-3)-(2-3)(2-0)=-2	CW	B
			E(3,0)	(2-0)(0-3)-(2-3)(3-0)=-3	CW	B
			F(0,0)	(2-0)(0-3)-(2-3)(0-0)=-6	CW	B

Signature _____

Page No. ☐

Date

3	B(2,2)	A(0,3)	C(1,1)	(0-2)(1-2) - (3-2)(1-2) + 3	CCW	C
		C(1,1)	D(2,1)	(1-2)(1-2) - (1-2)(2-2) + 1	CCW	D
		D(2,1)	E(3,0)	(2-2)(0-2) - (1-2)(3-2) + 1	CCW	E
4	E(3,0)	A(0,3)	B(2,2)	(0-3)(2-0) - (3-0)(2-3) - 3	CW	A
			C(1,1)	(0-3)(1-0) - (3-0)(1-3) + 3	CCW	C
		C(1,1)	D(2,1)	(1-3)(1-0) - (1-0)(2-3) - 1	CW	C
			F(0,0)	(1-3)(0-0) - (1-0)(0-3) + 3	CCW	F

Convex Hull: F(0,0) → E(3,0) → B(2,2) → A(0,3) → F(0,0)

Q2: Brute force Algorithm

```
n = len(P)
for i in 0 -- n-1:
    F[i] = 0
    for K in range(j, 0, -1):
        if P[0:K] == P[i-K+1 : i+1]:
            F[i] = K
            break
```

Time complexity: $O(n^3)$

2. KMP Optimized Algorithm:

```
n = len(P)
A = [0]*n
K = 0
for i in 1 -- n-1
    while K>0 and P[K] != P[i]:
```


Signature

Date

```
K = A[K-1]
if P[K] == P[i]
    K = K+1
    A[i] = K
```

while computing A[i], KMP uses the previously computed A[i-1] to avoid rechecking prefixes from scratch.

Time Complexity : $O(n)$

3. Computation of A[i] for "ababaca" using both approaches Brute force.

i	P[0...i]	K values.	A[i]
0	a	No proper Prefix → A[0]=0	0
1	ab	K=1; "a" vs "b" → no → A[1]=0	0
2	aba	K=2; "ab" vs "ba" → no; K=1; "a" vs "a" → match	1
3	ababa	K=3; "aba" vs "bab" → no; K=2; "ab" vs "ab" → match	2
4	ababa	K=4; "abab" vs "baba" → no; K=3; "aba" vs "aba" → match	3
5	ababac	K=5; "ababa" vs "babac" → no; K=4; "abab" vs "abac" → no; K=3: "aba" vs "bac" → no; K=2; "ab" vs "ac" → no; K=1 "a" vs "c" → no	0
6	ababaca	K=6; "ababac" vs "babaca"; K=5; "ababa" vs "abaca" → no; K=4: "abab" vs "baca" → no; K=3; "aba" vs "aca" → no; K=2; "ab" vs "ca" → no; K=1: "a" vs "a" → match	1

F = [0, 0, 1, 2, 3, 0, 1]

Signature

Date

Optimized KMP:

i	P[i]	K before	Compare (P[K] ?= P[i])	Action	K after	A[i]
0	a	-	-	F[0] = 0	-	0
1	b	K=F[0]=0	K=0 → 'a' ?= 'b' → no	K = 0	0	0
2	a	K=F[1]=0	P[0] ?= P[2] : 'a' ?= 'a' → Y	K = 1	1	1
3	b	K=F[2]=1	P[1] ?= P[3] : 'b' ?= 'b' → Y	K = 2	2	2
4	a	K=A[3]=2	P[2] ?= P[4] : 'a' ?= 'a' → Y	K = 3	3	3
5	c	K=A[4]=3	P[3] ?= P[5] : 'b' ?= 'c'	K=1 ; P[1]	K=0; P[0] ?=	0
			→ no → K = F[2] = 1	?= P[5] :	P[5] : 'a'	
				'b' ?= 'c' →	a ?= 'c' →	
				no → K=0	no → K=0	
6	a	K=A[5]=0	P[0] ?= P[6] : 'a' ?= 'a' → Yes	K = 1	1	1

F = [0, 0, 1, 2, 3, 0, 1]

4. Comparision.

(i) Time Complexity:

Brute force: $O(n^3)$

Optimized KMP: $O(n)$

(ii) Character Comparisions:

Brute force algorithm does many repeated comparisons of the same prefixes and suffixes from the beginning. KMP reuses previous meet matches and ensures each character is compared only a limited number of times.

Signature

Date

Q3:

(a) S = {1, 5, 6, 8} Change = 13 dp[13] = 8

i	dp[i] before	Coin = 1 dp[i] (aft)	Coin = 5 dp[i] (aft)	Coin = 6 dp[i] (aft)	Coin = 8 dp[i] (aft)
0	1	1	1	1	1
1	0	1	1	1	1
2	0	1	1	1	1
3	0	1	1	1	1
4	0	1	1	1	1
5	0	1	2	2	2
6	0	1	2	3	3
7	0	1	2	3	3
8	0	1	2	3	4
9	0	1	2	3	4
10	0	1	3	4	5
11	0	1	3	5	6
12	0	1	3	6	7
13	0	1	2	6	8

(b) str1 = KITTEN , str2 = SITTING

if str1[i-1] == str2[j-1]

dp[i][j] = dp[i-1][j-1]

else : delete insert substitution.

dp[i][j] = 1 + min(dp[i-1][j], dp[i][j-1], dp[i-1][j-1])

j	0	1	2	3	4	5	6	7
	ε	S	I	T	T	I	N	G

Signature

Date

	j	0	1	2	3	4	5	6	7
		''	S	I	T	T	I	N	G
i									
0	''	0	1	2	3	4	5	6	7
1	K	1	1	2	3	4	5	6	7
2	I	2	2	1	2	3	4	5	6
3	T	3	3	2	1	2	3	4	5
4	T	4	4	3	2	1	2	3	4
5	E	5	5	4	3	2	2	3	4
6	N	6	6	5	4	3	3	2	3

dp[6][7] = 3
Substitute K → S
Substitute E → I
Insert G at the end.

c). length[] = {1,2,3,4,5,6,7,8}, price = {1,5,8,9,10,16,18,20}
Rod length = 8

j	i : P_i + r[j-i]	r[j]	cut[j]
1	1 : $P_1 + r_0 \Rightarrow 1+0 \Rightarrow 1$	1	1
2	1 : 1+1 ⇒ 2 ; 2 : 5+0 ⇒ 5	5	2
3	1 : 1+5 → 6 ; 2 : 5+1 → 6 ; 3 : 8+0 → 8	8	3
4	1 : 1+8 → 9 ; 2 : 5+5 ⇒ 10 ; 3 : 8+9 → 9 ; 4 : 9+0 → 9	10	2
5	1 : 1+10 → 11 ; 2 : 5+8 → 13 ; 3 : 8+5 → 13 ; 4 : 9+1 → 10 ; 5 : 10+0 = 10	13	2
6	1 : 1+13 → 14 ; 2 : 5+10 → 15 ; 3 : 8+8 → 16 ; 4 : 9+5 → 14 ; 5 : 10+1 ⇒ 11 ; 6 : 16+0 ⇒ 16	16	3
7	1 : 1+16 ⇒ 17 ; 2 : 5+13 → 18 ; 3 : 8+10 → 18 ; 4 : 9+8 → 17 ; 5 : 10+5 → 15 ; 6 : 16+1 → 17 ; 7 : 18+0 → 18	18	2
8	1 : 1+18 → 19 ; 2 : 5+16 → 21 ; 3 : 8+13 → 21 ; 4 : 9+10 → 19 ; 5 : 10+8 → 18 ; 6 : 16+5 → 21 ; 7 : 18+1 → 19 ; 8 : 20+0 → 20	21	2

Signature _____

Date

r = [0, 1, 5, 8, 10, 13, 16, 18, 21]

cut = [0, 1, 2, 3, 2, 2, 3, 2, 2]

j = 8 → cut [8] = 2 ; 8-2 → 6

j = 6 → cut [6] = 3 ; 6-3 → 3

j = 3 → cut [3] = 3 ; 3-3 → 0

Total Value = P2 + P3 + P3 ⇒ 5 + 8 + 8 ⇒ 21

(d) word dict = {i, like, ice, cream, icecream, mobile, apple}

input = ilike apple.

j	dp [j]	Substring	Valid split	Valid Segmentation.
0	True	" "	base case	
1	True	i	" i "	" i like apple "
2	False	il	-	
3	False	ili	-	
4	False	ilik	-	
5	True	ilike	" i like "	
6	False	ilikea	-	
7	False	ilikeap	-	
8	False	ilikeapp	-	
9	False	ilikeappl	-	
10	True	ilikeapple	" i like apple "	

Q4: Scenario : Meal Planning with an Budget.

Goal: Maximize total nutrition while staying with a $15 weekly food budget

Weight: Cost (money) Capacity 15 dollars.

Value: Nutrition gained

Signature

Date

Data Table

Meals	Cost ($)	Nutrition (Value)
Salad	4	8
Chicken Rice	6	12
Pasta	5	10
Soup	3	6
Smoothie	2	5

dp [Meal] [budget]

Meal ↓ / Budget →	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Salad (4,8)	0	0	0	0	8	8	8	8	8	8	8	8	8	8	8	8
Chicken Rice (6,12)	0	0	0	0	8	8	12	12	12	12	20	20	20	20	20	20
Pasta (5,10)	0	0	0	0	8	10	12	12	12	18	20	22	22	22	22	22
Soup (3,6)	0	0	0	6	8	10	12	14	16	18	18	22	24	24	24	26
Smoothie (2,5)	0	0	5	6	8	10	12	14	17	19	21	23	25	26	27	27

Final: dp [5] [15] = 27

Max Value = 27

Best Meals: Chicken Rice + Pasta + soup + Smoothie.

Total Cost: 6 + 5 + 3 + 2 = 15

Nutrition: 27

SHOKIA

Signature