**Tutorial 9: Student Name\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Student id:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Question 1: Let X=aabbacab and Y=baabcbb. Find the shortest common super-sequence for X and Y. (Backtracking process is required.)

Solution:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Y j=0 | b j=1 | a j=2 | a | b | c | b | b |
| X i=0 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| a i=1 | 1 |  |  |  |  |  |  |  |
| a i=2 | 2 |  |  |  |  |  |  |  |
| b | 3 |  |  |  |  |  |  |  |
| b | 4 |  |  |  |  |  |  |  |
| a | 5 |  |  |  |  |  |  |  |
| c | 6 |  |  |  |  |  |  |  |
| a | 7 |  |  |  |  |  |  |  |
| b | 8 |  |  |  |  |  |  |  |

Backtracking:

Question 2:

**Input:**  An array A[1..n] of n integers (positive or negative).

**Task:**  Use dynamic method to find a non-empty interval [i, j] such that A[i]+A[i+1]+…+A[j] is maximized.

Example: Given an array: -1, 2 -3, 4, 5, -1

The sum of interval [1,1]=-1, [1,2]=-1+2=1, [3, 5]=-3+4+5=6.

Hint: Let d[i] be the cost of the max sum of intervals ending at position i.

That is, d[i]=max {sum[1,i], sum[2, i], …, sum[I,i]}.

Find recursive equation and use it to design a DP algorithm.

The final solution is the subinterval with the maximal d value.

In this example, d[1]=sum of [1,1]=-1, d[2]=sum of [2,2]=2.

d[3]=sum of [2,3]=-1, d[4]=sum of [4,4]=4.

Answer:

Alg:

Phase 1

d(1):=A[i]

For i=2 to n do:

If d(i-1)>0,

d(i)=d(i-1)+A[i], B[i]=1,

/\* containing A[i] and optimal interval ending at i-1.

Otherwise, d(i)=A[i], B[i]=0,

/\* the optimal interval ending at i contains only A[i].

//\* B for backtracking.

Phase 2: Find j with the maximal d value. (It can also be done in phase 1)

Phase 3: Backtracking:

i=j,

while(i>1 & B(i)=1)

j=j-1,

The optimal interval is [A[i],…, A[j]].