Tutorial 9:

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	↑ 2	↖ 2	↖ 3	↑ 4	↑ 5	↑ 6	↑ 7
a i=2	↑ 2	↑ 3	↖ 3	↖ 3	↑ 4	↑ 5	↑ 6	↑ 7
b	↑ 3	↖ 3	↑ 4	↑ 4	↖ 4	← 5	↖ 6	↖ 7
b	↑ 4	↖ 4	↑ 5	↑ 5	↖ 5	← 6	↖ 6	↖ 7
a	↑ 5	↑ 5	↖ 5	↖ 6	↑ 6	← 7	↑ 7	↑ 8
c	↑ 6	↑ 6	↑ 6	↑ 7	↑ 7	↖ 7	↑ 8	↑ 9
a	↑ 7	↑ 7	↖ 7	↖ 7	↑ 8	↑ 8	↑ 9	↑ 10
b	↑ 8	↖ 8	↑ 8	↑ 8	↖ 8	↑ 9	↖ 9	↖ 10

Backtracking: $b \rightarrow a \rightarrow c \rightarrow a \rightarrow b \rightarrow c \rightarrow b \rightarrow a \rightarrow a \rightarrow b$

$$SCS = \{baabcbacab\}$$

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:

$$d(i) = \begin{cases} A[i] \quad if\ i = 1 \\ d[i-1] + A[i]\ if\ d[i-1] > 0 \\ A[i] \quad if\ d[i-1] \leq 0 \end{cases}$$

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]].