جواب سوال اول به راحتی با کمک recursion-tree قابل اثبات است:
طارح :

```
fib(n){
    if(n==0 || n==1){
        return 1;
    else
    return fib(n-1) + fib(n-2);
}
```

$$\Rightarrow \quad T(n) = \begin{cases} 1 & ; n=0,1 \\ T(n-1)+T(n-1) & ; n>1 \end{cases} \Rightarrow$$

$$T(n) = T(n-1)+T(n-2)+C \;;$$

$n \qquad \Rightarrow n$

$n-1 \qquad n-2 \qquad \Rightarrow 2n$

$n-2 \qquad n-3 \qquad n-3 \qquad n-4 \qquad \Rightarrow 4n$

$\vdots$

(1)

$$T(n) = 2^0 n + 2^1 n + 2^2 n + \dots =$$

$$n(2^0+2^1+2^2+\dots) \approx 2^n$$

$$\Rightarrow O(2^n)$$

جواب سوال چهارم

$$P_1 \times P_2 \times P_3 \times P_4 \times P_5 \times P_6$$

$$5\times 10 \quad 10\times 3 \quad 3\times 12 \quad 12\times 5 \quad 5\times 50 \quad 50\times 6$$

$$D_0 \quad D_1 \quad D_2 \quad D_3 \quad D_4 \quad D_5 \quad D_6$$

$$m_{11} = m_{22} = m_{33} = m_{44} = m_{55} = m_{66} = 0$$

$$m_{12} = \min(m_{11} + m_{22} + d_0 d_1 d_2) = 0 + 0 + 5 \times 10 \times 3 = 150$$
$$m_{23} = \min(m_{22} + m_{33} + d_1 d_2 d_3) = 0 + 0 + 10 \times 3 \times 12 = 360$$
$$m_{34} = \min(m_{33} + m_{44} + d_2 d_3 d_4) = 0 + 0 + 3 \times 12 \times 5 = 180$$
$$m_{45} = \min(m_{44} + m_{55} + d_3 d_4 d_5) = 0 + 0 + 12 \times 5 \times 50 = 3000$$
$$m_{56} = \min(m_{55} + m_{66} + d_4 d_5 d_6) = 0 + 0 + 5 \times 50 \times 6 = 1500$$

$$m_{13} = \begin{cases} k=1 \Rightarrow \min(m_{11} + m_{23} + d_0 d_1 d_3) = 0 + 360 + 5 \times 10 \times 12 = 960 \\ k=2 \Rightarrow \min(m_{12} + m_{33} + d_0 d_2 d_3) = 150 + 0 + 5 \times 3 \times 12 = 330 \end{cases}$$

$$m_{24} = \begin{cases} k=2 \Rightarrow \min(m_{22} + m_{34} + d_1 d_2 d_4) = 0 + 180 + 10 \times 3 \times 5 = 330 \\ k=3 \Rightarrow \min(m_{23} + m_{44} + d_1 d_3 d_4) = 360 + 0 + 10 \times 12 \times 5 = 960 \end{cases}$$

$$m_{35} = \begin{cases} k=3 \Rightarrow \min(m_{33} + m_{45} + d_2 d_3 d_5) = 0 + 3000 + 3 \times 12 \times 50 = 4800 \\ k=4 \Rightarrow \min(m_{34} + m_{55} + d_2 d_4 d_5) = 180 + 0 + 3 \times 5 \times 50 = 930 \end{cases}$$

$$m_{46} = \begin{cases} k=4 \Rightarrow \min(m_{44} + m_{56} + d_3 d_4 d_6) = 0 + 1500 + 12 \times 5 \times 6 = 1860 \\ k=5 \Rightarrow \min(m_{45} + m_{66} + d_3 d_5 d_6) = 3000 + 0 + 12 \times 50 \times 6 = 3600 \end{cases}$$

$$m_{14} = \begin{cases} k=1 \Rightarrow \min(m_{11} + m_{24} + d_0 d_1 d_4) = 0 + 330 + 5 \times 10 \times 5 = 580 \\ k=2 \Rightarrow \min(m_{12} + m_{34} + d_0 d_2 d_4) = 150 + 180 + 5 \times 10 \times 5 = 580 \\ k=3 \Rightarrow \min(m_{13} + m_{44} + d_0 d_3 d_4) = 330 + 0 + 5 \times 12 \times 5 = 630 \end{cases}$$

$$m_{25} = \begin{cases} k=2 \Rightarrow \min(m_{22} + m_{35} + d_1 d_2 d_5) = 0 + 930 + 10 \times 12 \times 50 = 6930 \\ k=3 \Rightarrow \min(m_{23} + m_{45} + d_1 d_3 d_5) = 360 + 3000 + 10 \times 12 \times 50 = 9360 \\ k=4 \Rightarrow \min(m_{24} + m_{55} + d_1 d_4 d_5) = 330 + 0 + 10 \times 5 \times 50 = 2830 \end{cases}$$

$$m_{36} = \begin{cases} k=3 \Rightarrow \min(m_{33} + m_{46} + d_2 d_3 d_6) = 0 + 3600 + 3 \times 12 \times 6 = 3816 \\ k=4 \Rightarrow \min(m_{34} + m_{56} + d_2 d_4 d_6) = 180 + 1500 + 3 \times 5 \times 6 = 1770 \\ k=5 \Rightarrow \min(m_{35} + m_{66} + d_2 d_5 d_6) = 930 + 0 + 3 \times 50 \times 6 = 1830 \end{cases}$$

$$m_{15} = \begin{cases} k=1 = \min(m_{11}+m_{25}+d_0 d_1 d_5) = 0+2830+5\times10\times50 = 5330 \\ k=2 = \min(m_{12}+m_{35}+d_0 d_2 d_5) = 150+930+5\times3\times50 = 1830 \\ k=3 = \min(m_{13}+m_{45}+d_0 d_3 d_5) = 150+930+5\times12\times50 = 1880 \\ k=4 = \min(m_{14}+m_{55}+d_0 d_4 d_5) = 580+0+5\times5\times50 = 1830 \end{cases}$$

$$m_{26} \begin{cases} k=2 = \min(m_{22}+m_{36}+d_1 d_2 d_6) = 0+1770+10\times3\times6 = 1950 \\ k=3 = \min(m_{23}+m_{46}+d_1 d_3 d_6) = 360+3400+10\times12\times6 = \text{[illegible]} \\ k=4 = \min(m_{24}+m_{56}+d_1 d_4 d_6) = 330+1500+10\times5\times6 = 2130 \\ k=5 = \min(m_{25}+m_{66}+d_1 d_5 d_6) = 2830+0+10\times50\times6 = 4480 \\ \qquad 5840 \end{cases}$$

$$m_{16} = \begin{cases} k=1 = \min(m_{11}+m_{26}+d_0 d_1 d_6) = 0+1950+5\times10\times6 = 2250 \\ k=2 = \min(m_{12}+m_{36}+d_0 d_2 d_6) = 150+1770+5\times3\times6 = 2010 \\ k=3 = \min(m_{13}+m_{46}+d_0 d_3 d_6) = 330+3400+5\times12\times6 = 4090 \\ k=4 = \min(m_{14}+m_{56}+d_0 d_4 d_6) = 580+1500+5\times5\times6 = 2230 \\ k=5 = \min(m_{15}+m_{66}+d_0 d_5 d_6) = 1830+0+5\times50\times6 = 3330 \end{cases}$$

تشکیل ماتریس

	۱	۲	۳	۴	۵	۶
۱	۰	۱۵۰	۳۳۰	۵۸۰	۱۸۳۰	۲۰۱۰
۲		۰	۳۶۰	۳۳۰	۲۸۳۰	۱۹۵۰
۳			۰	۱۸۰	۹۳۰	۱۷۷۰
۴				۰	۳۰۰۰	۳۴۰۰
۵					۰	۱۵۰۰
۶						۰

⇓

تشکیل ماتریس P

	۱	۲	۳	۴	۵	۶
۱		۱	۲	۱	۲	۲
۲			۲	۲	۴	۲
۳				۳	۴	۴
۴					۴	۵
۵						۵
۶						

$\xrightarrow[\text{ضرب شدن ماتریس ها}]{\text{تعیین نحوه}}$ $(P_1 P_2)(P_3 P_4 P_5 P_6)$

(۶ و ۳) ماتریس

(۳ و ۶)

$\Longrightarrow (P_1 \times P_2) \times ((P_3 \times P_4) \times (P_5 \times P_6))$ ⇐ حداقل ۵ ضرب لازم است.

جواب سوال ۶) یک ماتریس در نظر می‌گیریم و سطر و ستون آن را پر می‌کنیم:

x_i \ y_j		1	0	0	1	0	1	0	1
	۰	۰	۰	۰	۰	۰	۰	۰	۰
0	۰	↖۰	↖۱	↖۱	←۱	↖۱	←۱	↖۱	←۱
1	۰	↖۱	←۱	←۱	↖۲	←۲	↑۲	←۲	↑۲
0	۰	↑۱	↖۲	↖۲	←۲	↖۳	←۳	↖۳	←۳
1	۰	↖۱	↑۲	←۲	↖۳	←۳	↖۴	←۴	↖۴
1	۰	↖۱	↑۲	←۲	↖۳	←۳	↖۴	←۴	↖۵
0	۰	↑۱	↖۲	↖۳	←۳	↑۴	←۴	↑۵	←۵
1	۰	↖۱	↑۲	↑۳	↖۴	←۴	↖۵	←۵	↖۶
1	۰	↖۱	↑۲	↑۳	↖۴	↑۴	↖۵	↑۵	↖۶
0	۰	↑۱	↖۲	↖۳	↑۴	↖۵	↑۵	↖۶	←۶

"C"

$$c[i,j] = \begin{cases} 0 & ; \; i=0 \;||\; j=0 \\ c[i-1,j-1]+1 & ; \; i,j>0 \;\&\; x_i = y_j \\ \max(c[i,j-1], c[i-1,j]) & ; \; i,j>0 \;\&\; x_i \neq y_j \end{cases}$$

حال طبق ماتریس longest commen subsequence را بدست می‌آوریم:

طول lcs ← ۶ می‌باشد:

LCS : < 1, 0, 0, 1, 1, 0 >

جواب سوال ۸)

```
Print-Lcs (c, x, Y)
n = c[x.length, Y.length]
let S[1...n] be a new array
i = x.length
j = Y.length
while i>0 && j>0
   if x[i] == Y[j]
      S[n] = x[i]
      n = n-1
      i = i-1
      j = j-1
   else if c[i-1, j] >= c[i, j-1]
      i = i-1
   else
      j = j-1
```

```
for (i=0 to S.length)
   print S[i]
```

ادامه صفحه بعد

```
Lsc - length (X, Y, c, b)

    m = |X|
    n = |Y|
    if c [m, n] != 0 || m == 0 || n == 0
        return
    if X [m] == Y [n]
        b [m, n] = '↖'
        c [m, n] = Lcs-length(X[1 ...m-1], Y[1...n-1], c, b)+1
    else if Lcs-length(X[1 ...m-1, Y, c, b) >=
        Lcs-length(X, Y[1... n-1], c, b)
        b [m, n] = '↑'
        c [m, n] = Lcs-length (X [1 ...m-1, Y, c, b)
    else
        b [m, n] = '←'
        c [m, n] = Lcs-length (X, Y[1...n-1], c, b)


memo-lcs-length (X, Y)

    let c [1...|X|, 1 ...|Y|] && b[1...|X|, 1...|Y|] be
        a new tables
    memo-lcs-length (X, Y, c, b)
    return c and b
```

جواب سوال ۹)

```
optimal-BST(root, i, j, last)
    if i == j
        return
    if last == 0
        print root[i, j] + "is the root"
    else if    j < last:
        print root[i, j] + "is the left child of" + last
    else
        print root[i, j] + "is the right child of" + last
    optimal-BST(root, i, root[i, j]-1, root[i, j])
    optimal-BST(root, root[i, j]+1, j, root[i, j])
```

جواب سوال ۱۵) ماتریس مربوطه را تشکیل می دهیم:

با استفاده از فرمول زیر ماتریس را می نویسیم و ماتریس P را تشکیل می دهیم:

$$A[i][j] = \min_{i \le k \le j}(A[i, k-1] + A[k+1, j]) + \sum_{m=i}^{j} P_m \qquad i < j$$

به دلیل زیاد بودن محاسبات اینجا ننوشتم:

	0	1	2	3	4	5	6	7	8
1	0								
2		0	۰/۰۴						
3			0	۰/۰۷					
4				0	۰/۰۸				
5					0	۰/۰۷			
6						0	۰/۱		
7							0	۰/۰۱۳	
8								0	۰/۱۴
9									0

ادامه صفحه بعد

در نهایت بعد از تشکیل ماتریس ها و از روی آن خواهیم داشت :