Algorithm 2021 Spring HW3

(Chapter 9 and Chapter 15)

1. (25pts) There are two sorted arrays, each containing n numbers. Please give an algorithm to find the median of all $2n$ elements, and it's running time should be bounded by $O(\lg n)$.

Ans：

Two_sorted_arrays_median (A, B, n)

```
1.  if (n == 1)
2.      return min(A[0], B[0])
3.  else if (n % 2 == 1)
4.      median_index ← n % 2 + 1
5.      else median_index ← n % 2
6.      compare A[median_index] and B[median_index]
7.      a ← take the lower part of the array with the greater median
8.      b ← take the upper part of the array with the lesser median
9.      Two_sorted_arrays_median (a, b, ⌊n/2⌋)
```

2. (25pts) Determine the cost of an optimal binary search tree for a set of n = 5 keys with the following probabilities:

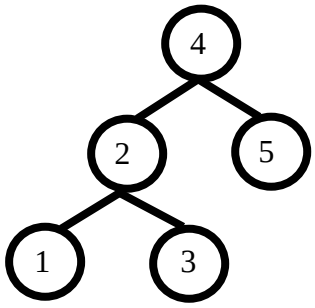
i	0	1	2	3	4	5
p_i		0.08	0.15	0.05	0.1	0.12
q_i	0.04	0.1	0.08	0.1	0.06	0.12

(A) 2.85
(B) 1.98
(C) 2.12
(D) 1.92
(note: level starts at 1.)
(note2: need to construct the tree)
Ans: A

W	5	4	3	2	1	0
1	0.04	0.1	0.08	0.1	0.06	0.12
2	0.22	0.33	0.23	0.26	0.3	
3	0.45	0.48	0.39	0.5		
4	0.6	0.64	0.63			
5	0.76	0.88				
6	1					

C	1	2	3	4	5	6
0	0.04	0.1	0.08	0.1	0.06	0.12
1	0.36	0.51	0.41	0.42	0.48	
2	0.89	0.99	0.86	1.04		
3	1.37	1.57	1.52			
4	1.98	2.35				
5	2.85					

Tree



3. (25pts)

```
LCS-Length(X, Y, m, n)
1  for i ← 1 to m do
2      c[i, 0] ← 0
3  for j ← 1 to n do
4      c[0, j] ← 0
5  for i ← 1 to m do
6      for j ← 1 to n do
7          if x_i = y_j then
8              c[i, j] ← c[i − 1, j − 1] + 1
9              b[i, j] ← " ↖ "
10         else if c[i − 1, j] ≥ c[i, j − 1] then
11             c[i, j] ← c[i − 1, j]
12             b[i, j] ← " ↑ "
13         else
14             c[i, j] ← c[i, j − 1]
15             b[i, j] ← " ← "
16 return c and b
```

Please find the longest common subsequence of string “ACBDSABBE” and “CADDSBEAB” by using the above algorithm. You need to draw the table to show your process.

Ans：

		0	1	2	3	4	5	6	7	8	9
		j	C	A	D	D	S	B	E	A	B
0	*i*	X 0	X 0	X 0	X 0	X 0	X 0	X 0	X 0	X 0	X 0
1	A	X 0	↑ 0	↖ 1	←1	←1	←1	←1	←1	↖ 1	←1
2	C	X 0	↖ 1	↑ 1	↑ 1	↑ 1	↑ 1	↑ 1	↑ 1	↑ 1	↑ 1
3	B	X 0	↑ 1	↑ 1	↑ 1	↑ 1	↑ 1	↖ 2	← 2	← 2	↖ 2
4	D	X 0	↑ 1	↑ 1	↖ 2	↖ 2	← 2	↑ 2	↑ 2	↑ 2	↑ 2
5	S	X 0	↑ 1	↑ 1	↑ 2	↑ 2	↖ 3	← 3	← 3	← 3	← 3
6	A	X 0	↑ 1	↖ 2	↑ 2	↑ 2	↑ 3	↑ 3	↑ 3	↖ 4	← 4
7	B	X 0	↑ 1	↑ 2	↑ 2	↑ 2	↑ 3	↖ 4	← 4	↑ 4	↖ 5
8	B	X 0	↑ 1	↑ 2	↑ 2	↑ 2	↑ 3	↖ 4	↑ 4	↑ 4	↖ 5
9	E	X 0	↑ 1	↑ 2	↑ 2	↑ 2	↑ 3	↑ 4	↖ 5	← 5	↑ 5

We found longest common subsequence “ADSAB” with length 5.

4. (25pts) Matrix-chain

Consider the multiplication of five matrices $A_1A_2A_3A_4A_5$, and the dimensions are as follows:

matrix	A_1	A_2	A_3	A_4	A_5
dimension	20×25	25×15	15×10	10×20	20×30

Please find the minimum number of scalar multiplications, optimal parenthesization and fill in the two tables.

Ans：

20750

((A1(A2A3))(A4A5))

Table m:

J	1	2	3	4	5	I
	0	7500	8750	12750	20750	1
		0	3750	8750	16250	2
			0	3000	10500	3
				0	6000	4
					0	5

Table s:

J	2	3	4	5	I
	1	1	3	3	1
		2	3	3	2
			3	3	3
				4	4

If i==j then m[i, j] = 0

m[1, 2] = min{
m[1, 1] + m[2, 2] + p0p1p2 = 0 + 0 + 7500 = 7500
}
s[1, 2] = 1 because m[1, 1] + m[2, 2] + p0p1p2 = 0 + 0 + 7500 = 7500 is min.
m[2, 3] = min{
m[2, 2] + m[3, 3] + p1p2p3 = 0 + 0 + 3750 = 3750
}
s[2, 3] = 2 because m[2, 2] + m[3, 3] + p1p2p3 = 0 + 0 + 3750 = 3750 is min.
m[3, 4] = min{
m[3, 3] + m[4, 4] + p2p3p4 = 0 + 0 + 3000 = 3000
}
s[3, 4] = 3 because m[3, 3] + m[4, 4] + p2p3p4 = 0 + 0 + 3000 = 3000 is min.
m[4, 5] = min{
m[4, 4] + m[5, 5] + p3p4p5 = 0 + 0 + 6000 = 6000
}
s[4, 5] = 4 because m[4, 4] + m[5, 5] + p3p4p5 = 0 + 0 + 6000 = 6000 is min.
m[1, 3] = min{
m[1, 1] + m[2, 3] + p0p1p3 = 0 + 3750 + 5000 = 8750
m[1, 2] + m[3, 3] + p0p2p3 = 7500 + 0 + 3000 = 10500
}
s[1, 3] = 1 because m[1, 1] + m[2, 3] + p0p1p3 = 0 + 3750 + 5000 = 8750 is min.
m[2, 4] = min{
m[2, 2] + m[3, 4] + p1p2p4 = 0 + 3000 + 7500 = 10500
m[2, 3] + m[4, 4] + p1p3p4 = 3750 + 0 + 5000 = 8750
}
s[2, 4] = 3 because m[2, 3] + m[4, 4] + p1p3p4 = 3750 + 0 + 5000 = 8750 is min.
m[3, 5] = min{
m[3, 3] + m[4, 5] + p2p3p5 = 0 + 6000 + 4500 = 10500
m[3, 4] + m[5, 5] + p2p4p5 = 3000 + 0 + 9000 = 12000
}
s[3, 5] = 3 because m[3, 3] + m[4, 5] + p2p3p5 = 0 + 6000 + 4500 = 10500 is min.
m[1, 4] = min{
m[1, 1] + m[2, 4] + p0p1p4 = 0 + 8750 + 10000 = 18750
m[1, 2] + m[3, 4] + p0p2p4 = 7500 + 3000 + 6000 = 16500
m[1, 3] + m[4, 4] + p0p3p4 = 8750 + 0 + 4000 = 12750
}

s[1, 4] = 3 because m[1, 3] + m[4, 4] + p0p3p4 = 8750 + 0 + 4000 = 12750 is min.
m[2, 5] = min{
m[2, 2] + m[3, 5] + p1p2p5 = 0 + 10500 + 11250 = 21750
m[2, 3] + m[4, 5] + p1p3p5 = 3750 + 6000 + 7500 = 17250
m[2, 4] + m[5, 5] + p1p4p5 = 8750 + 0 + 15000 = 23750
}
s[2, 5] = 3 because m[2, 3] + m[4, 5] + p1p3p5 = 3750 + 6000 + 7500 = 17250 is min.
m[1, 5] = min{
m[1, 1] + m[2, 5] + p0p1p5 = 0 + 17250 + 15000 = 32250
m[1, 2] + m[3, 5] + p0p2p5 = 7500 + 10500 + 9000 = 27000
m[1, 3] + m[4, 5] + p0p3p5 = 8750 + 6000 + 6000 = 20750
m[1, 4] + m[5, 5] + p0p4p5 = 12750 + 0 + 12000 = 24750
}
s[1, 5] = 3 because m[1, 3] + m[4, 5] + p0p3p5 = 8750 + 6000 + 6000 = 20750 is min.