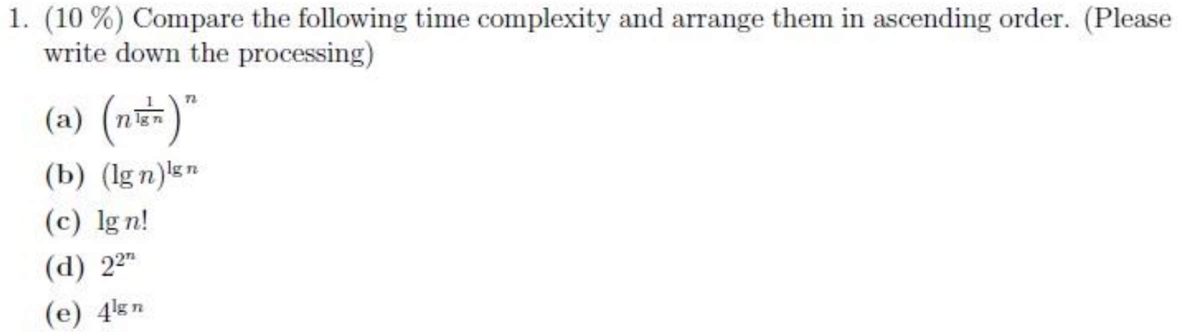
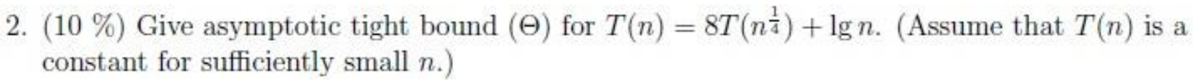
Algorithm Spring 2021 Midterm Exam



1. (10 %) Compare the following time complexity and arrange them in ascending order. (Please write down the processing)
   (a) $\left(n^{\frac{1}{\lg n}}\right)^n$
   (b) $(\lg n)^{\lg n}$
   (c) $\lg n!$
   (d) 2^{2^n}
   (e) $4^{\lg n}$

Ans：c < e < b < a < d



2. (10 %) Give asymptotic tight bound (Θ) for $T(n) = 8T(n^{\frac{1}{4}}) + \lg n$. (Assume that $T(n)$ is a constant for sufficiently small n.)

Ans：$\theta((lgn)^{\frac{3}{2}})$

Let $n = 2^m$

⇨ $T(2^m) = 8\,T\left(2^{\frac{m}{4}}\right) + lg2^m$

$= 8\,T\left(2^{\frac{m}{4}}\right) + m$

Let $T(2^m) = S(m)$

⇨ $S(m) = 8\,S(\frac{m}{4}) + m$

Using master theorem 1,

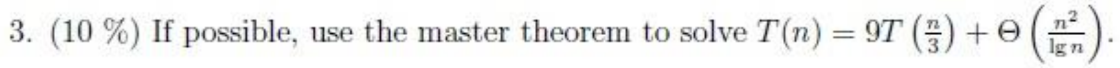
⇨ $f(m) = m = O\left(m^{\log_4 8-\varepsilon}\right)\ for\ \ \varepsilon = \log_4 8 - 1 = \frac{3}{2} - 1 = \frac{1}{2} > 0$

⇨ $S(m) = \theta\left(m^{\log_4 8}\right) = \theta(m^{\frac{3}{2}})$

Because $n = 2^m,\ m = lgn,$

Therefore, $S(m) = T(2^m) = \theta(m^{\frac{3}{2}})$

⇨ $T(n) = \theta((lgn)^{\frac{3}{2}})$



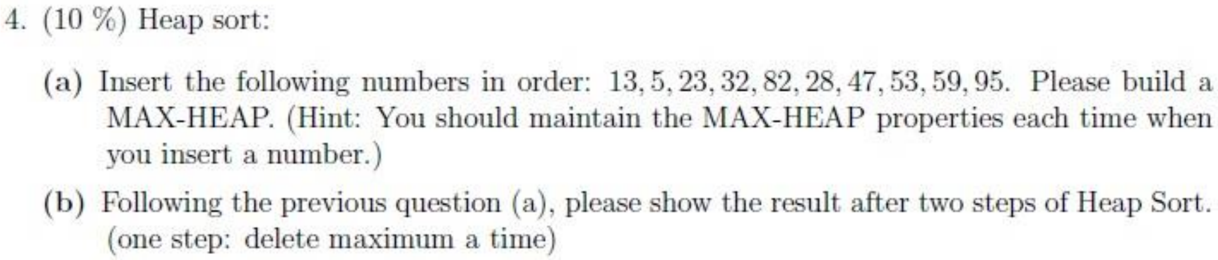
3. (10 %) If possible, use the master theorem to solve $T(n) = 9T\left(\frac{n}{3}\right) + \Theta\left(\frac{n^2}{\lg n}\right)$.

Ans：

$By\ Extended\ Master\ Theorem$:

$$If\ f(n) = \Theta(n^{\log_b a}(\log_b n)^{-1}), then\ T(n) = \Theta(n^{\log_b a} \log_b \log_b n)$$

$$\begin{aligned} f(n) &= \Theta(n^2(\log_2 n)^{-1}) \\ &= \Theta(n^{\log_3 9}(\log_3 n)^{-1}) \rightarrow T(n) = \Theta(n^{\log_3 9} \log_3 \log_3 n) \\ &= \Theta(n^2 \log_3 \log_3 n) \\ &= \Theta(n^2 loglogn) \end{aligned}$$



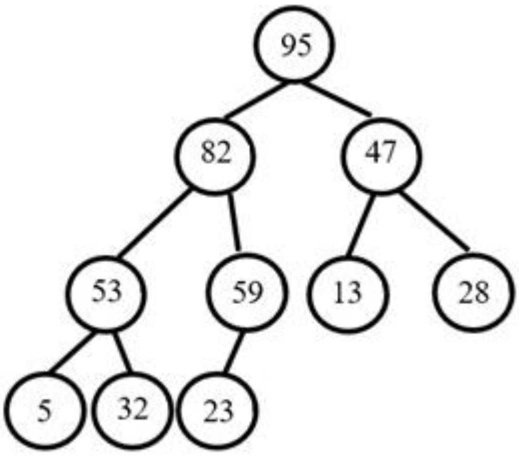
4. (10 %) Heap sort:

(a) Insert the following numbers in order: 13, 5, 23, 32, 82, 28, 47, 53, 59, 95. Please build a MAX-HEAP. (Hint: You should maintain the MAX-HEAP properties each time when you insert a number.)

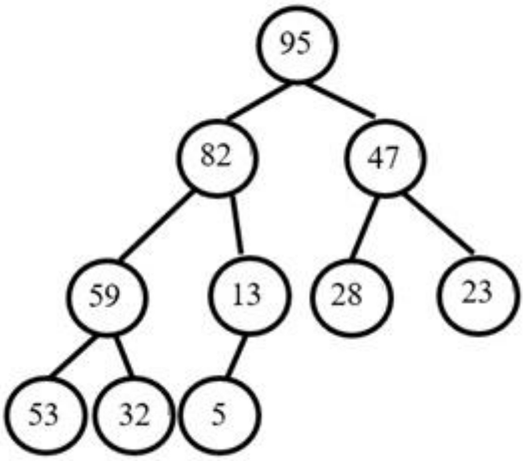
(b) Following the previous question (a), please show the result after two steps of Heap Sort. (one step: delete maximum a time)

(a)

Insert one by one：

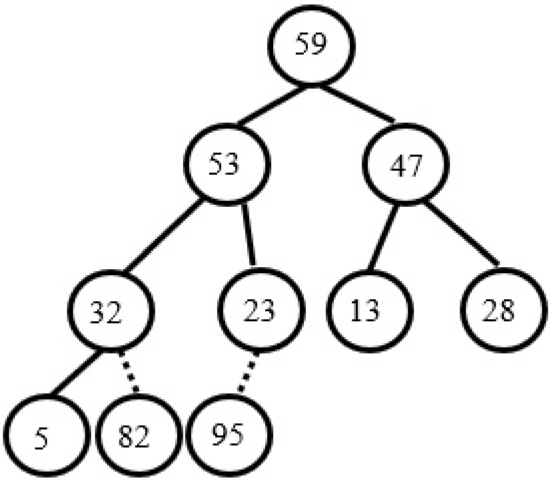


BUILD-MAX-HEAP algo.：

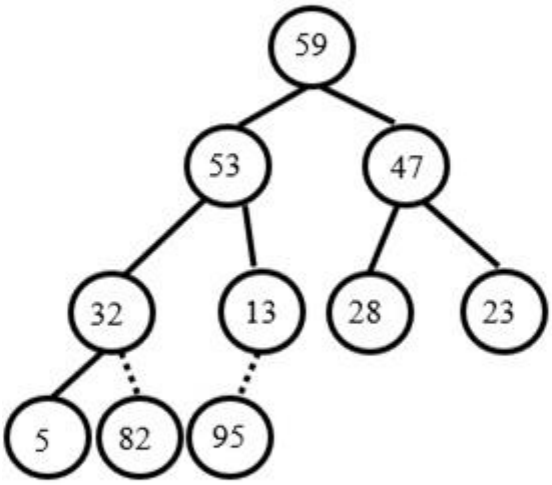


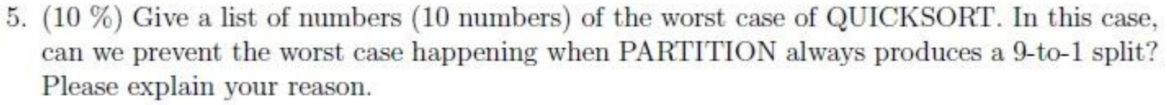
(b)

Insert one by one：



BUILD-MAX-HEAP algo.：

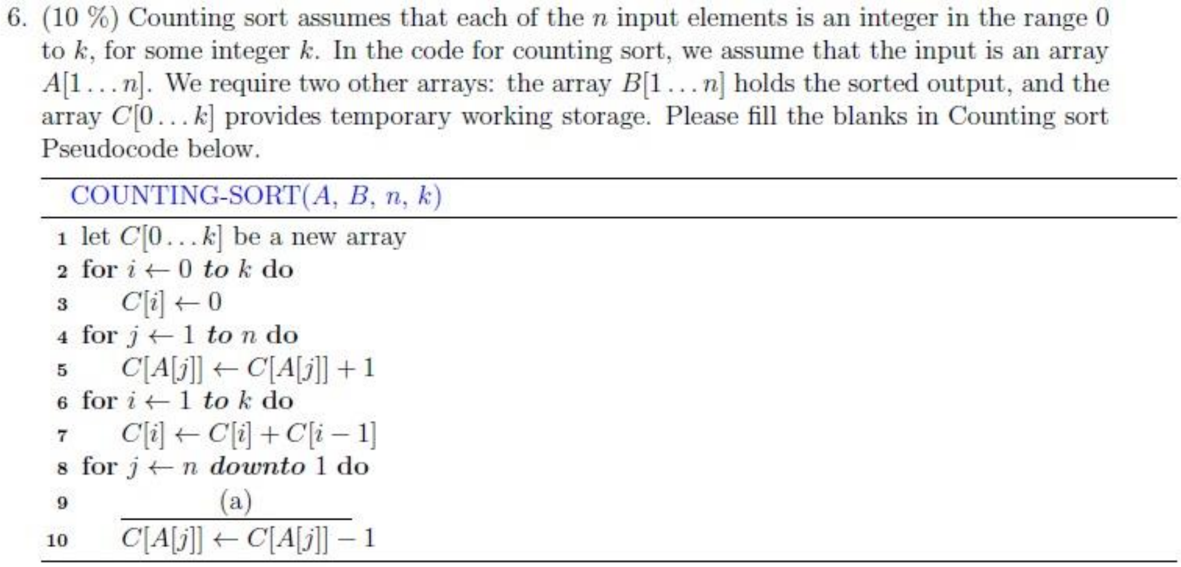




5. (10 %) Give a list of numbers (10 numbers) of the worst case of QUICKSORT. In this case, can we prevent the worst case happening when PARTITION always produces a 9-to-1 split? Please explain your reason.

Ans: (1) 9, 8, 7, 6, 5, 4, 3, 2, 1, 0.

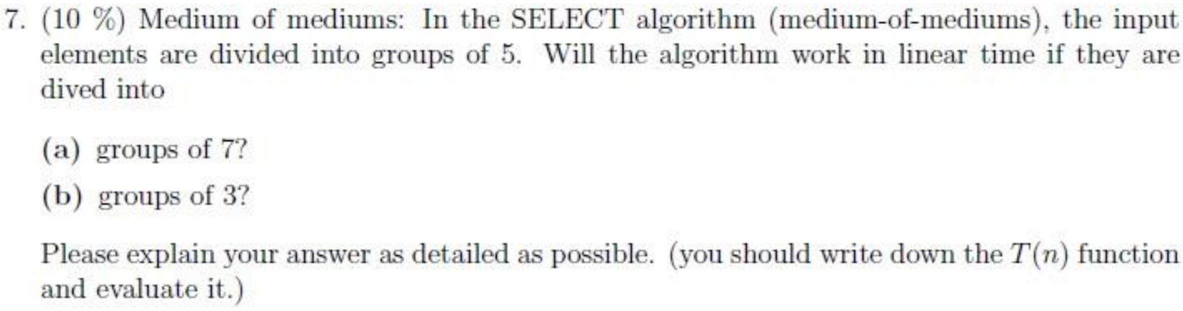
(2) No, because producing 9-to-1 split always partition the number one by one. Therefore, we can't prevent the worst case happened.



6. (10 %) Counting sort assumes that each of the n input elements is an integer in the range 0 to k, for some integer k. In the code for counting sort, we assume that the input is an array $A[1 \ldots n]$. We require two other arrays: the array $B[1 \ldots n]$ holds the sorted output, and the array $C[0 \ldots k]$ provides temporary working storage. Please fill the blanks in Counting sort Pseudocode below.

```
COUNTING-SORT($A$, $B$, $n$, $k$)
let $C[0 \ldots k]$ be a new array
for $i \leftarrow 0$ to $k$ do
    $C[i] \leftarrow 0$
for $j \leftarrow 1$ to $n$ do
    $C[A[j]] \leftarrow C[A[j]] + 1$
for $i \leftarrow 1$ to $k$ do
    $C[i] \leftarrow C[i] + C[i-1]$
for $j \leftarrow n$ downto 1 do
    ________(a)________
    $C[A[j]] \leftarrow C[A[j]] - 1$
```

Ans: $B\left[C\left[A[j]\right]\right] \leftarrow A[j]$



7. (10 %) Medium of mediums: In the SELECT algorithm (medium-of-mediums), the input elements are divided into groups of 5. Will the algorithm work in linear time if they are dived into

(a) groups of 7?

(b) groups of 3?

Please explain your answer as detailed as possible. (you should write down the $T(n)$ function and evaluate it.)

in case 5

Step 1：Divide the n elements into groups of 5. Get $\lceil n/5 \rceil$ groups.

Step 2：Find the median of each of the $\lceil n/5 \rceil$ groups.

Step 3：Find the median x of the $\lceil n/5 \rceil$ medians by a recursive call to SELECT.

Step 4：Using the modified version of PARTITION that takes the pivot element as input, partition the input array around x.

Step 5：Consider the worst case.

Step 1: making groups of 5 elements takes $O(n)$ time.

Step 2: sorting $\lceil n/5 \rceil$ groups in $O(1)$ time each.

Step 3 takes time $T(\lceil n/5 \rceil)$

Step 4: partitioning the n-element array around x takes $O(n)$ time.

Step 5 takes time $\leq T(7n/10+6)$, assuming that $T(n)$ is monotonically increasing.

$$T(n) \leq \begin{cases} O(1) & \text{if } n < 140 \\ T(\lceil n/5 \rceil) + T(7n/10+6) + O(n) & \text{if } n \geq 140 \end{cases}$$

Ans：

(a) Yes, $T(n) \leq T\left(\left\lceil \frac{n}{7} \right\rceil\right) + T\left(\frac{5n}{7}+8\right) + O(n) \rightarrow \theta(n)$ analyze by recurrent tree

$\because$ $T(n) \leq \theta(n)$ $\therefore$ $T(n)$ will be bounded by linear time.

(b) No, $T(n) \leq T\left(\left\lceil \frac{n}{3} \right\rceil\right) + T\left(\frac{2n}{3}+4\right) + O(n) \rightarrow \theta(n \log n)$ analyze by recurrent tree $\because$ $T(n) \leq \theta(n \log n)$ $\therefore$ $T(n)$ will not be bounded by linear time in worst case.

8. (10 %) Determine the minimum expected search cost and an optimal binary search tree for a set of $n = 6$ keys with the following probabilities:

i	1	2	3	4	5	6
p_i	0.1	0.25	0.15	0.1	0.15	0.25

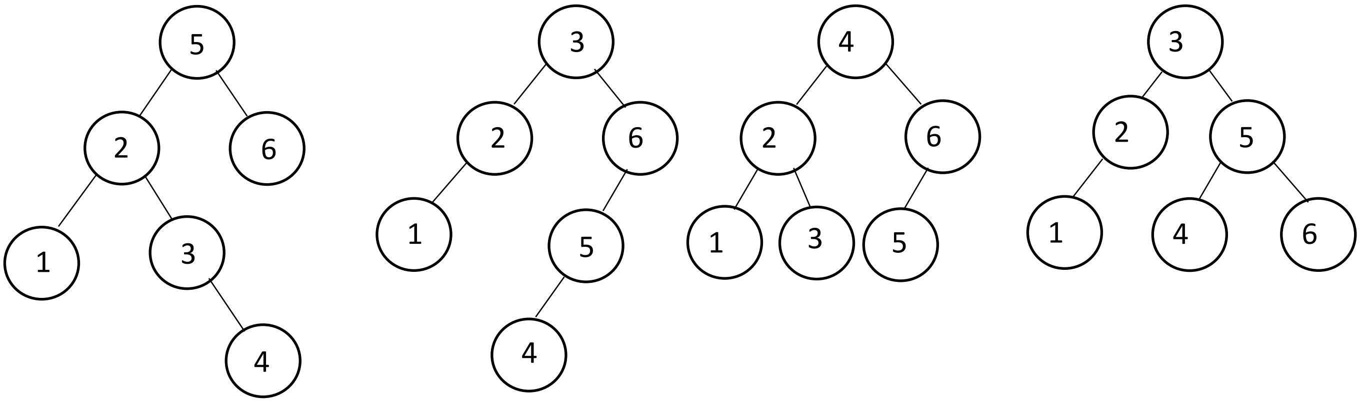
where $k_1 < k_2 < k_3 < k_4 < k_5 < k_6$ and p_i is the probabilities of k_i.

Ans：minimum search cost = 2.3

w	j							
i		0	1	2	3	4	5	6
	1	0	0.1	0.35	0.5	0.6	0.75	1
	2		0	0.25	0.4	0.5	0.65	0.9
	3			0	0.15	0.25	0.4	0.65
	4				0	0.1	0.25	0.5
	5					0	0.15	0.4
	6						0	0.25
	7							0

e	j							
i		0	1	2	3	4	5	6
	1	0	0.1	0.45	0.75	1.05	1.55	2.3
	2		0	0.25	0.55	0.85	1.25	2
	3			0	0.15	0.35	0.7	1.25
	4				0	0.1	0.35	0.85
	5					0	0.15	0.55
	6						0	0.25
	7							0

Trees:



9. (10 %) Find the minimum number of scalar multiplications and an optimal parenthesization of a matrix-chain product whose sequence of dimensions is $\langle 3, 50, 5, 12, 6, 10, 5\rangle$.

Ans：1476 (((((A1A2)A3)A4)A5)A6)

Table m:

J	1	2	3	4	5	6	I
	0	750	930	1146	1326	1476	1
		0	3000	1860	3160	2060	2
			0	360	660	810	3
				0	720	660	4
					0	300	5
						0	6

Table s:

J	2	3	4	5	6	I
	1	2	3	4	5	1
		2	2	2	2	2
			3	4	4	3
				4	4	4
					5	5

m[1, 2] = min{

m[1, 1] + m[2, 2] + p0p1p2 = 0 + 0 + 750 = 750

}

s[1, 2] = 1 because m[1, 1] + m[2, 2] + p0p1p2 = 0 + 0 + 750 = 750 is min.

m[2, 3] = min{

m[2, 2] + m[3, 3] + p1p2p3 = 0 + 0 + 3000 = 3000

}

s[2, 3] = 2 because m[2, 2] + m[3, 3] + p1p2p3 = 0 + 0 + 3000 = 3000 is min.

m[3, 4] = min{

m[3, 3] + m[4, 4] + p2p3p4 = 0 + 0 + 360 = 360

}

s[3, 4] = 3 because m[3, 3] + m[4, 4] + p2p3p4 = 0 + 0 + 360 = 360 is min.

m[4, 5] = min{

m[4, 4] + m[5, 5] + p3p4p5 = 0 + 0 + 720 = 720

}

s[4, 5] = 4 because m[4, 4] + m[5, 5] + p3p4p5 = 0 + 0 + 720 = 720 is min.

m[5, 6] = min{

m[5, 5] + m[6, 6] + p4p5p6 = 0 + 0 + 300 = 300

}

s[5, 6] = 5 because m[5, 5] + m[6, 6] + p4p5p6 = 0 + 0 + 300 = 300 is min.

m[1, 3] = min{

m[1, 1] + m[2, 3] + p0p1p3 = 0 + 3000 + 1800 = 4800

m[1, 2] + m[3, 3] + p0p2p3 = 750 + 0 + 180 = 930

}

s[1, 3] = 2 because m[1, 2] + m[3, 3] + p0p2p3 = 750 + 0 + 180 = 930 is min.

m[2, 4] = min{

m[2, 2] + m[3, 4] + p1p2p4 = 0 + 360 + 1500 = 1860

m[2, 3] + m[4, 4] + p1p3p4 = 3000 + 0 + 3600 = 6600

}

s[2, 4] = 2 because m[2, 2] + m[3, 4] + p1p2p4 = 0 + 360 + 1500 = 1860 is min.

m[3, 5] = min{

m[3, 3] + m[4, 5] + p2p3p5 = 0 + 720 + 600 = 1320

m[3, 4] + m[5, 5] + p2p4p5 = 360 + 0 + 300 = 660

}

s[3, 5] = 4 because m[3, 4] + m[5, 5] + p2p4p5 = 360 + 0 + 300 = 660 is min.

m[4, 6] = min{

m[4, 4] + m[5, 6] + p3p4p6 = 0 + 300 + 360 = 660

m[4, 5] + m[6, 6] + p3p5p6 = 720 + 0 + 600 = 1320

}

s[4, 6] = 4 because m[4, 4] + m[5, 6] + p3p4p6 = 0 + 300 + 360 = 660 is min.

m[1, 4] = min{

m[1, 1] + m[2, 4] + p0p1p4 = 0 + 1860 + 900 = 2760

m[1, 2] + m[3, 4] + p0p2p4 = 750 + 360 + 90 = 1200

m[1, 3] + m[4, 4] + p0p3p4 = 930 + 0 + 216 = 1146

}

s[1, 4] = 3 because m[1, 3] + m[4, 4] + p0p3p4 = 930 + 0 + 216 = 1146 is min.

m[2, 5] = min{

m[2, 2] + m[3, 5] + p1p2p5 = 0 + 660 + 2500 = 3160

m[2, 3] + m[4, 5] + p1p3p5 = 3000 + 720 + 6000 = 9720 m[2, 4] + m[5, 5] + p1p4p5 = 1860 + 0 + 3000 = 4860

}

s[2, 5] = 2 because m[2, 2] + m[3, 5] + p1p2p5 = 0 + 660 + 2500 = 3160 is min.

m[3, 6] = min{

m[3, 3] + m[4, 6] + p2p3p6 = 0 + 660 + 300 = 960

m[3, 4] + m[5, 6] + p2p4p6 = 360 + 300 + 150 = 810

m[3, 5] + m[6, 6] + p2p5p6 = 660 + 0 + 250 = 910

}

s[3, 6] = 4 because m[3, 4] + m[5, 6] + p2p4p6 = 360 + 300 + 150 = 810 is min.

m[1, 5] = min{

m[1, 1] + m[2, 5] + p0p1p5 = 0 + 3160 + 1500 = 4660

m[1, 2] + m[3, 5] + p0p2p5 = 750 + 660 + 150 = 1560

m[1, 3] + m[4, 5] + p0p3p5 = 930 + 720 + 360 = 2010

m[1, 4] + m[5, 5] + p0p4p5 = 1146 + 0 + 180 = 1326

}

s[1, 5] = 4 because m[1, 4] + m[5, 5] + p0p4p5 = 1146 + 0 + 180 = 1326 is min.

m[2, 6] = min{

m[2, 2] + m[3, 6] + p1p2p6 = 0 + 810 + 1250 = 2060

m[2, 3] + m[4, 6] + p1p3p6 = 3000 + 660 + 3000 = 6660

m[2, 4] + m[5, 6] + p1p4p6 = 1860 + 300 + 1500 = 3660

m[2, 5] + m[6, 6] + p1p5p6 = 3160 + 0 + 2500 = 5660

}

s[2, 6] = 2 because m[2, 2] + m[3, 6] + p1p2p6 = 0 + 810 + 1250 = 2060 is min.

m[1, 6] = min{

m[1, 1] + m[2, 6] + p0p1p6 = 0 + 2060 + 750 = 2810

m[1, 2] + m[3, 6] + p0p2p6 = 750 + 810 + 75 = 1635

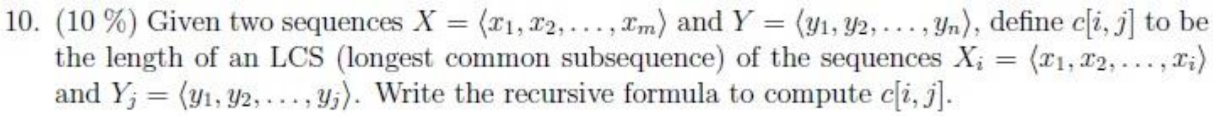
m[1, 3] + m[4, 6] + p0p3p6 = 930 + 660 + 180 = 1770

m[1, 4] + m[5, 6] + p0p4p6 = 1146 + 300 + 90 = 1536

m[1, 5] + m[6, 6] + p0p5p6 = 1326 + 0 + 150 = 1476

}

s[1, 6] = 5 because m[1, 5] + m[6, 6] + p0p5p6 = 1326 + 0 + 150 = 1476 is min.



10. (10 %) Given two sequences $X = \langle x_1, x_2, \ldots, x_m \rangle$ and $Y = \langle y_1, y_2, \ldots, y_n \rangle$, define $c[i, j]$ to be the length of an LCS (longest common subsequence) of the sequences $X_i = \langle x_1, x_2, \ldots, x_i \rangle$ and $Y_j = \langle y_1, y_2, \ldots, y_j \rangle$. Write the recursive formula to compute $c[i, j]$.

Ans：

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

or

```
$initial{:}\ c[i,\ j] = 0\ for\ i = 0\ or\ j = 0$
$function{:}$
    $if\ x_i = y_j\ then$
        $c[i,\ j] = c[i-1,\ j-1] + 1$
    $else\ if\ c[i-1,\ j] \geq c[i,\ j-1]\ then$
        $c[i,\ j] = c[i-1,\ j]$
    $else$
        $c[i,\ j] = c[i,\ j-1]$
```