Algorithm Middle Exam. 110.11.19

/* 答案紙 請寫上：學號 姓名 ；試卷共 6 頁 總分 100

1. (5%) What is the definition of Algorithm? (3 個條件)

2. (10%) Design a Turing to verify "Can 160 be divided by 16?"

3. (10%) 假設一程序如下，請求出其時間複雜度 O(T(n)) = ?

```
Procedure test(X, low, high)      /* X 為一陣列  */
  {
    mid = └(low+high)/2┘;   O(1)
  If (high > low) then {    O(1)
      test(X, low, mid);
      test(X, low, mid);     2T(n/2)
      test(X, mid, high);    2T(n/2)
      test(X, mid, high);
      a = b + c;
      }
  }
```

$O(k)$

$\sum_{h=0}^{\lfloor \lg n \rfloor} \left(\frac{n}{2^{h+1}}\right) = n \sum_{h=0}^{\lfloor \lg n \rfloor} \left(\frac{h}{2^h}\right)$

$\sum_{h=0}^{\infty} \left(\frac{h}{2^h}\right) = 2$

On

$O\left(n \sum_{h=0}^{\lfloor \lg n \rfloor} \left(\frac{h}{2^h}\right)\right)$

16
64

4. (10 %) 解 T(n) = 2 T(n/2) + n lg n

64
64
256
384
4096

5. (15%) Following procedure is a heap sort algorithm.

(1) (10%)求 BUILD-MAX-Heap(A) 時間複雜度？ n

4^6 $(2^2)^6$
4096 2^{12}

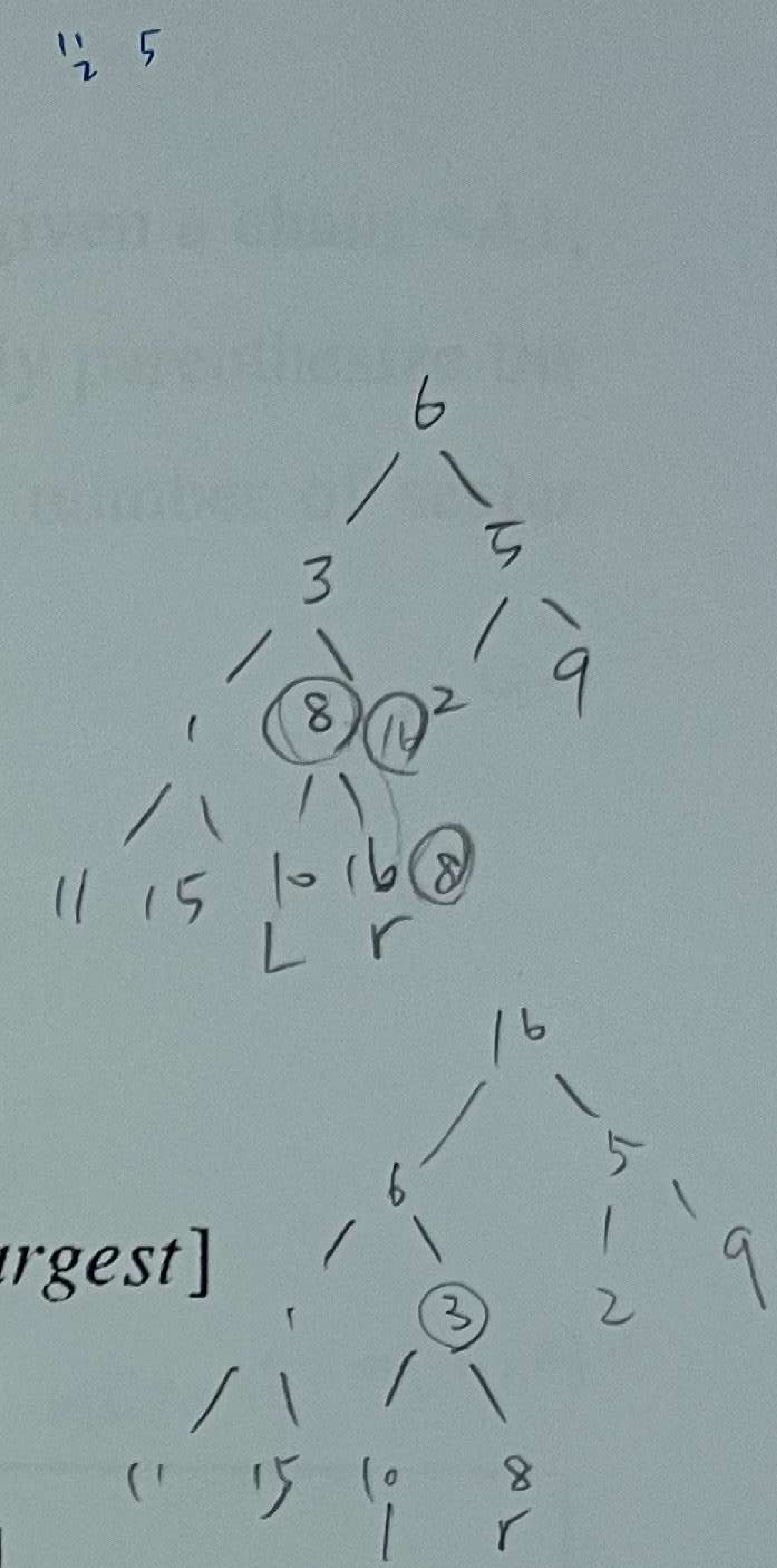
HEAPSORT(A, n)
BUILD-MAX-HEAP(A, n)
for $i \leftarrow n$ downto 2 n
  do exchange $A[1] \leftrightarrow A[i]$
    MAX-HEAPIFY($A, 1, i - 1$) lg n

2 2 2 2 2 2 64
16 32

```
BUILD-MAX-HEAP(A)
  heap-size[A] ← length[A]
  for i ← ⌊length[A]/2⌋ downto 1
        do MAX-HEAPIFY(A, i)
```

5

```
MAX-HEAPIFY(A, i)
  l ← LEFT(i)
  r ← RIGHT(i)
  if l ≤ heap-size[A] and A[l] > A[i]
     then largest ← l
     else largest ← i
  if r ≤ heap-size[A] and A[r] > A[largest]
     then largest ← r
  if largest ≠ i
     then exchange A[i] ↔ A[largest]
          MAX-HEAPIFY(A, largest)
```



6

(2) (5%) 若 A=[6, 3, 5, 1, 8, 2, 9, 11, 15, 10, 16]，求 Max –heap tree =? (每一步驟均要畫出)

6. (10%) The following algorithm is counting sort. If array A is following:

```
For i ← 0 to k        0 to 6
    Do C[i] = 0
For j ← 1 to n
    Do C[A[j]] ← C[A[j]] + 1
For i ← 1 to k
    Do C[i] = C[i] + C[i - 1]
For j ← n downto 1
    Do B[C[A[j]]] ← A[j]        B[C[2]]
       C[A[j]] ← C[A[j]] -1
```

Please write the initial state and each state of array A & B & C in the step of algorithm.

A[4, 6, 3, 2, 4, 2, 1, 2]

7. (20%) The matrix-chain multiplication problem is: given a chain <A1, A2, …, An> of n matrices, where for i =1, 2, …, n, fully parenthesize the product A1, A2, …, An in a way that minimizes the number of scalar multiplications.

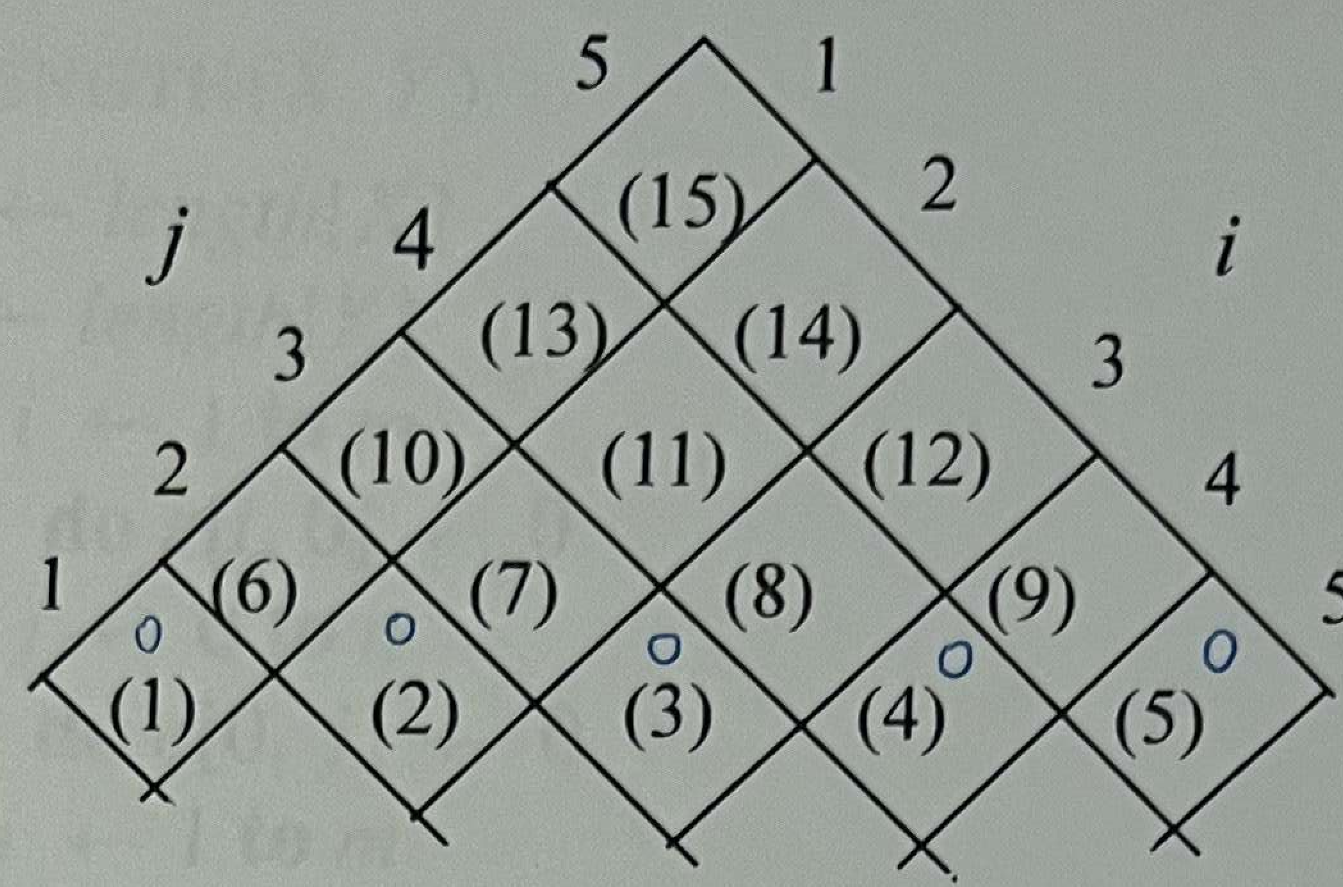
```
MATRIX-CHAIN-ORDER(p)
n ← length[p] − 1
for i ← 1 to n
    do m[i, i] ← 0
for l ← 2 to n                        ▷ l is the chain length.
    do for i ← 1 to n − l + 1
           do j ← i + l − 1
              m[i, j] ← ∞
              for k ← i to j − 1
                  do q ← m[i, k] + m[k + 1, j] + p_{i−1} p_k p_j
                     if q < m[i, j]
                        then m[i, j] ← q
                             s[i, j] ← k
return m and s
```

```
PRINT-OPTIMAL-PARENS(s, i, j)
if i = j
   then print "A"_i
   else print "("
        PRINT-OPTIMAL-PARENS(s, i, s[i, j])
        PRINT-OPTIMAL-PARENS(s, s[i, j] + 1, j)
        print ")"
```

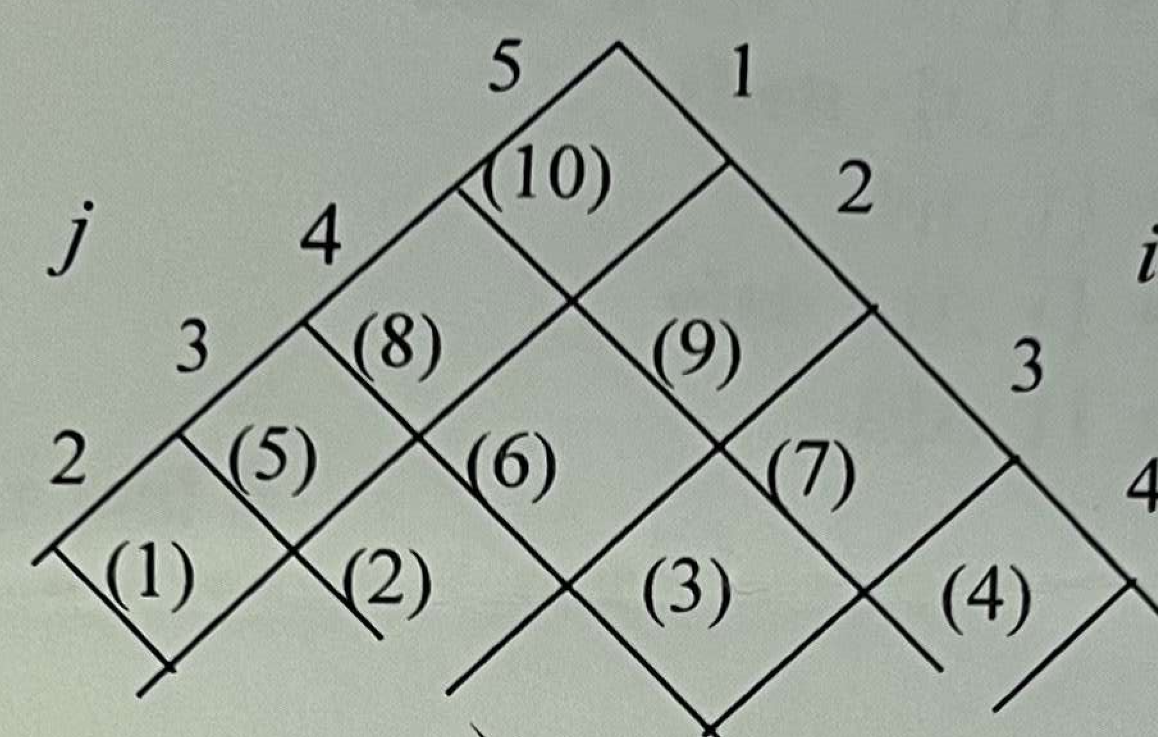
Data: A1(5, 5), A2(5, 10), A3(10, 5), A4(5, 20), A5(20, 10)

$A_1(A_2 A_3)$

m



s



(1) (5%) What is the purpose of line 8 and line 9 in MATRIX-CHAIN-ORDER(p)？(說明此兩行指令功用)

(2) (15%) 求 Matrix m(*i,j*) & s(*i,j*) (請依答案號次填入答案，每一表各 5%，(評分：錯誤：**1** 個，滿分；**2-4** 個，半對；**>4** 個，全錯)) & 最佳解(5%如何乘？最少乘法次數？)為何 ？

$(A_1\ A_2\ A_3)(A_4\ A_5)$

$(A_1\ (A_2\ A_3))($

8. (20%) Longest common subsequence

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

```
PRINT-LCS(b, X, i, j)
if i = 0 or j = 0
   then return
if b[i, j] = "↖"
   then PRINT-LCS(b, X, i − 1, j − 1)
        print x_i
elseif b[i, j] = "↑"
   then PRINT-LCS(b, X, i − 1, j)
else PRINT-LCS(b, X, i, j − 1)
```

Data: X={D, A, C, B, D, A} & Y={B, D, C, B, A}

(1) (5%)What is the purpose of line 10 and line 11 in LCS-LENGTH(X, Y) ?

(說明此兩行指令功用)

(2) 求 b[m, n] & c[m, n] (直接一起填入相同答號中) 及 最佳解(最長共同子序列？(必須以上述方法求出為真正答案))為何？(請依答案號次填入答案，每一表 5%，(評分：錯誤：1-3 個，滿分；4-6 個，半對；>6 個，全錯)) & 最佳解(5%最長共同子序列)？

b & c 表

		j	0	**1**	**2**	**3**	**4**	**5**
			Y_j	**B**	**D**	**C**	**B**	**A**
i	0	X_i	**(1)**	**(2)**	**(3)**	**(4)**	**(5)**	**(6)**
	1	D	**(7)**	**(8)**	**(9)**	**(10)**	**(11)**	**(12)**
	2	A	**(13)**	**(14)**	**(15)**	**(16)**	**(17)**	**(18)**
	3	C	**(19)**	**(20)**	**(21)**	**(22)**	**(23)**	**(24)**
	4	B	**(25)**	**(26)**	**(27)**	**(28)**	**(29)**	**(30)**
	5	D	**(31)**	**(32)**	**(33)**	**(34)**	**(35)**	**(36)**
	6	A	**(37)**	**(38)**	**(39)**	**(40)**	**(41)**	**(42)**

國立臺南大學　學年度　學期　考試　答案卷

總分 90

考試科目：演算法　　班別：數位三

學號：S10855[redacted]　　姓名：[redacted]

1、(1)有限的步驟 (2)有效率的運算 (3)有終止點

2、$160=(10100000)_2$

$16=(10000)_2$

$Q=\{q_0, q_1, q_2, q_3, q_4, q_N, q_Y\}$

$\Gamma=\{0, 1, b\}$

$I \subseteq \Gamma$

b: blank, in $\Gamma - I$

q_0: initial state

q_N: No state

q_Y: Yes state

	0	1	b
q_0	$(q_0, 0, R)$	$(q_0, 1, R)$	(q_1, b, L)
q_1	$(q_2, 0, L)$	$(q_N, 1, S)$	
q_2	$(q_3, 0, L)$	$(q_N, 1, S)$	(q_N, b, S)
q_3	$(q_4, 0, L)$	$(q_N, 1, S)$	(q_N, b, S)
q_4	$(q_Y, 0, S)$	$(q_N, 1, S)$	(q_N, b, S)

Yes, 160 can be divided by 16

過程：$(q_1, b, L) \to (q_2, 0, L) \to (q_3, 0, L) \to (q_4, 0, L) \to (q_Y, 0, S)$

3. $T(n)=\begin{cases} O(1) & \text{, if } n=1 \\ 4T(\frac{n}{2})+O(1) & \text{, if } n>1 \quad (n \geq 2) \end{cases}$

常數不計算

$T(n)=4T(\frac{n}{2})+1$　　令 $n=2^k$, $k=\lg n$

$=4[4T(\frac{n}{2^2})]$

$=4^2T(\frac{n}{2^2})$

$=4^2[4T(\frac{n}{2^3})]$

$=4^3T(\frac{n}{2^3})$

$= \vdots$

$=4^kT(\frac{n}{2^k})$

XS

$=2^{(2^k)}T(\frac{n}{2^k})$

$=2^nT(1)$

$=O(2^n)$

$T(n)=O(2^n)$

−10

$\boxed{4^k=(2^2)^k=2^{2k}}$

$T(n)=4T(\frac{n}{2})+1$

$=4[4T(\frac{n}{4})+1]+1$

$=4^2T(\frac{n}{2^2})+4+1$

$=\cdots$

$=4^kT(\frac{n}{2^k})+4^{k-1}+\ldots+1$

$=4^kT(1)+(4^{k+1}-1)/3$

$=n^2+\frac{4}{3n^2}-\frac{1}{3}$

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

4. $T(n) = 2T(\frac{n}{2}) + n\lg n$

令 $n = 2^k$, $k = \lg n$

$$T(n) = 2[2T(\frac{n}{2^2}) + \frac{n}{2}\lg\frac{n}{2}] + n\lg n$$
$$= 2^2T(\frac{n}{2^2}) + n\lg\frac{n}{2} + n\lg n$$
$$= 2^2[2T(\frac{n}{2^3}) + \frac{n}{2^2}\lg\frac{n}{2^2}] + n\lg\frac{n}{2} + n\lg n$$
$$= 2^3T(\frac{n}{2^3}) + n\lg\frac{n}{2^2} + n\lg\frac{n}{2} + n\lg n$$
$$= \vdots$$
$$= 2^kT(\frac{n}{2^k}) + n\sum_{i=0}^{k-1}(\lg\frac{n}{2^i})$$
$$= 2^kT(\frac{n}{2^k}) + n\sum_{i=0}^{k-1}(\lg n - i)$$
$$= nT(1) + n\sum_{i=0}^{k-1}(k-i)$$
$$= nT(1) + n(\frac{k^2+k}{2})$$
$$= nT(1) + \frac{1}{2}(n\lg^2 n + \lg n)$$
$$= O(n\lg^2 n)$$

5.

(1) BUILD-MAX-Heap(A) 的時間複雜度為 $O(n)$

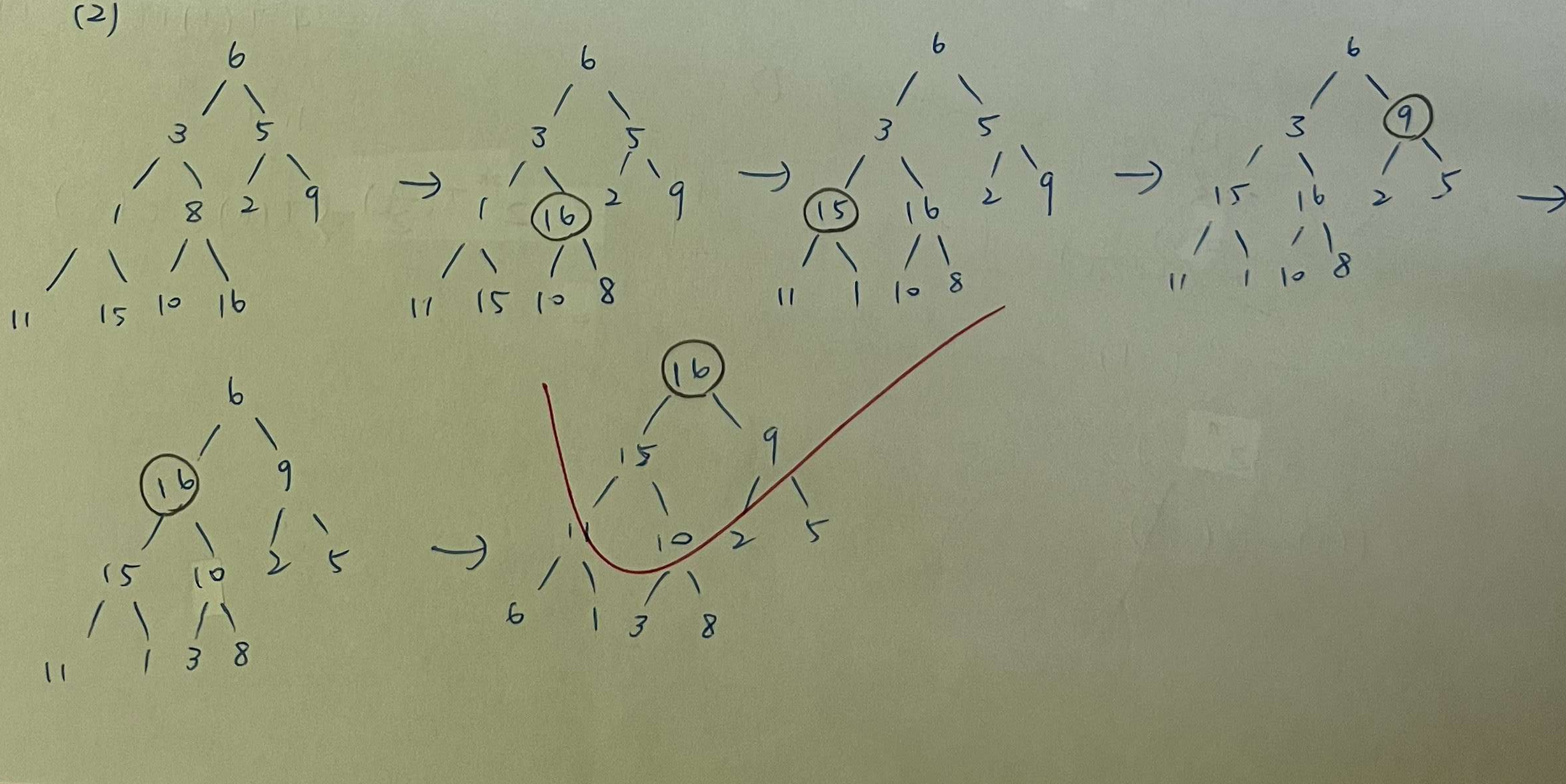
雖然 HAX-HEAPIFY 的時間複雜度為 $O(\lg n)$ 且執行了 $O(n)$ 次

但並不是所有的節點都有相同的深度，$O(n\lg n)$ 是被高估的

證： (h: 高度)

$$\sum_{h=0}^{\lfloor\lg n\rfloor}\lceil\frac{n}{2^{h+1}}\rceil O(h) = O(n\sum_{h=0}^{\lfloor\lg n\rfloor}\frac{h}{2^h})$$
$$\sum_{h=0}^{\infty}(\frac{h}{2^h}) = 2$$
$$O(n\sum_{h=0}^{\lfloor\lg n\rfloor}\frac{h}{2^h}) = O(n\sum_{h=0}^{\infty}\frac{h}{2^h}) = O(n)$$

(2)



6.

A | 4 | 6 | 3 | 2 | 4 | 2 | 1 | 2 |

C (0–6) | 0 | 0 | 0 | 0 | 0 | 0 | 0 | → C (0–6) | 0 | 1 | 3 | 1 | 2 | 0 | 1 | → C | 0 | 1 | 4 | 5 | 7 | 7 | 8 |

↙(由後往前,才能有stable特性)↘

→ A | 4 | 6 | 3 | 2 | 4 | 2 | 1 | 2 |

B | | | | 2 | | | | |

C (0–6) | 0 | 1 | 3 | 5 | 7 | 7 | 8 |

加入B後減1

→ A | 4 | 6 | 3 | 2 | 4 | 2 | 1 | 2 |

B | 1 | | | 2 | | | | |

C (0–6) | 0 | 0 | 3 | 5 | 7 | 7 | 8 |

→ …… →

→ A | 4 | 6 | 3 | 2 | 4 | 2 | 1 | 2 |

B | 1 | 2 | 2 | 2 | 3 | 4 | 4 | 6 |

C (0–6) | 0 | 0 | 1 | 4 | 5 | 7 | 7 |

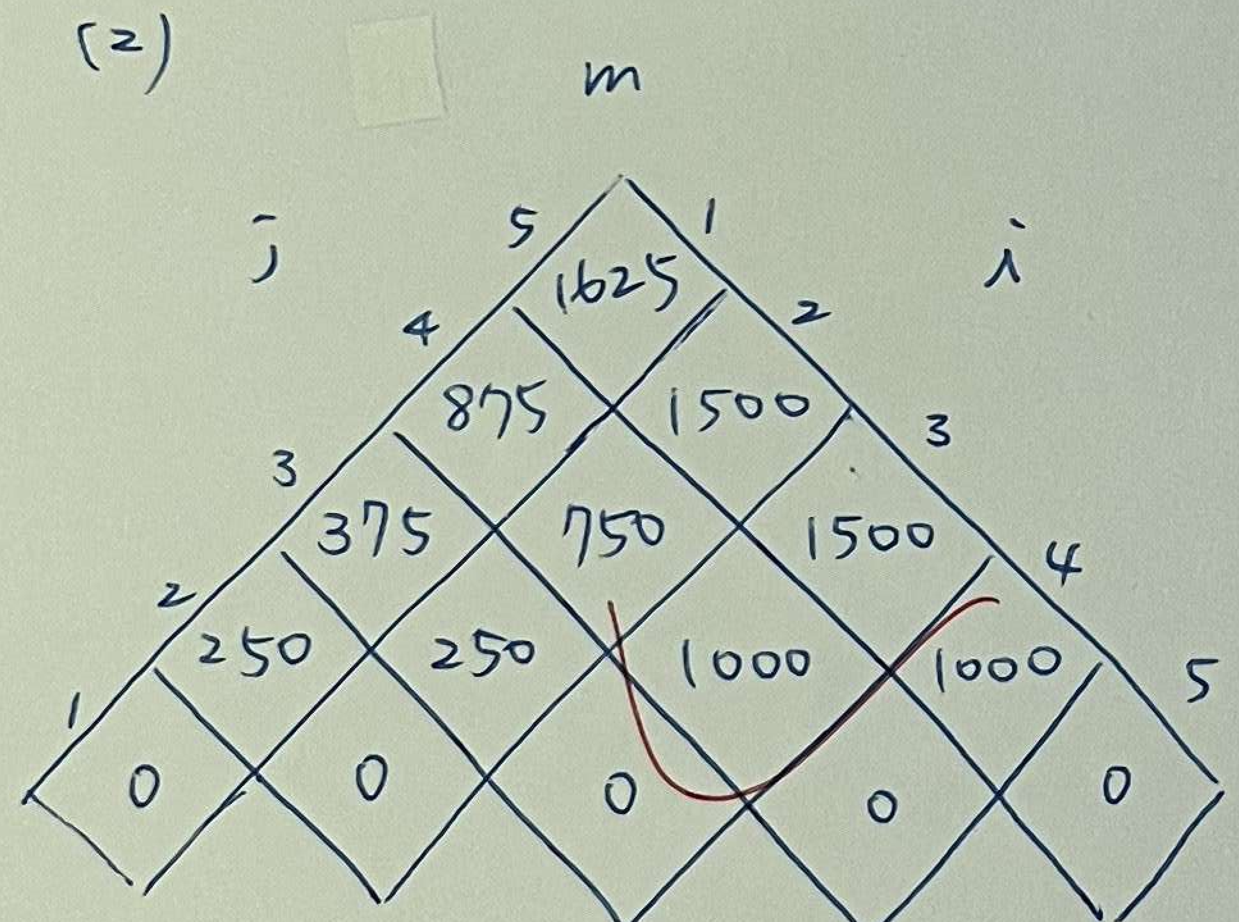
7. (1) 從 m[1,2] 開始,設k為切割處

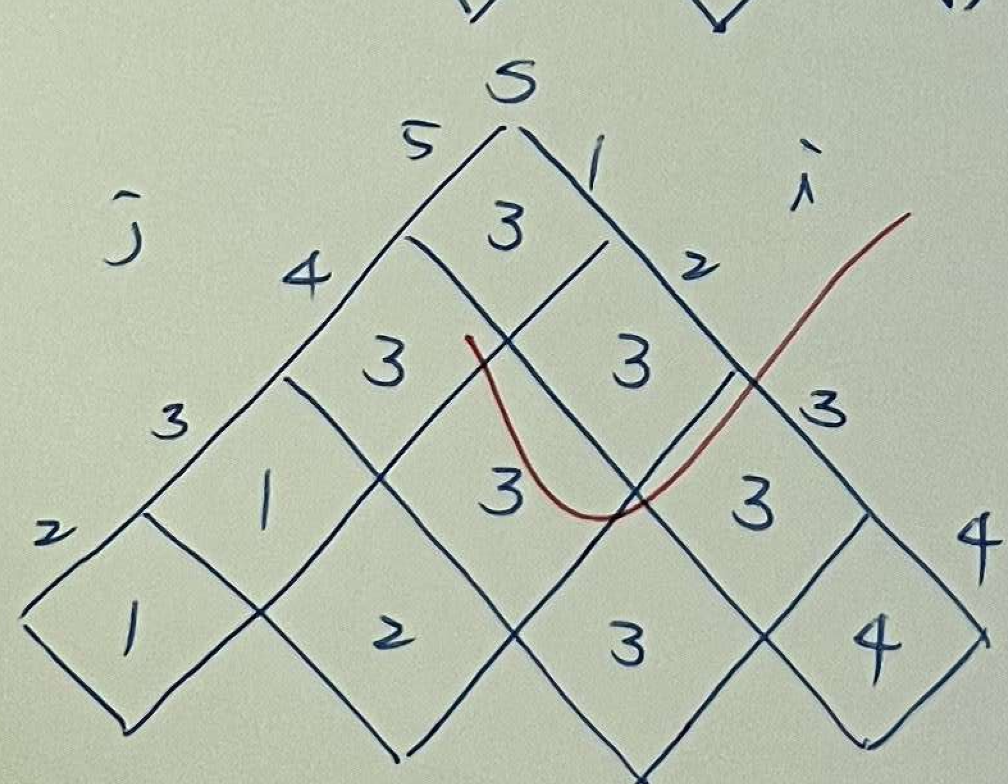
將 $m[1,k] + m[k+1,2] + \underbrace{P_{i-1}P_kP_j}_{\text{乘法次數}}$ 寫入 q

例: $A_1(5,5)$, $A_2(5,10)$, $A_3(10,5)$

則 $(A_1A_2)A_3$ 的 $P_{i-1}P_kP_j$ 為 $5\times10\times5$

(2)





最佳解: $(A_1(A_2A_3))(A_4A_5)$

最少乘法次數: 1625 次

$m[1,2] = 0+0+5\times5\times10 = 250$

$m[2,3] = 0+0+5\times10\times5 = 250$

$m[3,4] = 0+0+10\times5\times20 = 1000$

$m[4,5] = 0+0+5\times20\times10 = 1000$

$m[1,3] =$ 切1 $= m[1,1]+m[2,3]+5\times5\times5 = 375$

$=$ 切2 $= m[1,2]+m[3,3]+5\times10\times5 = 500$

$m[2,4] =$ 切2 $= m[2,2]+m[3,4]+5\times10\times20 = 2000$

$=$ 切3 $= m[2,3]+m[4,4]+5\times5\times20 = 750$

$m[3,5] =$ 切3 $= m[3,3]+m[4,5]+10\times5\times10 = 1500$

$=$ 切4 $= m[3,4]+m[5,5]+10\times20\times10 = 3000$

$m[1,4] =$ 切1 $= m[1,1]+m[2,4]+5\times5\times20 = 1250$

$=$ 切2 $= m[1,2]+m[3,4]+5\times10\times20 = 2250$

$=$ 切3 $= m[1,3]+m[4,4]+5\times5\times20 = 875$

$m[2,5] =$ 切2 $= m[2,2]+m[3,5]+5\times10\times10 = 2000$

$=$ 切3 $= m[2,3]+m[4,5]+5\times5\times10 = 1500$

$=$ 切4 $= m[2,4]+m[5,5]+5\times20\times10 = 1750$

↘

$m[1,5] = 切1 = m[1,1]+m[2,5] + 5\times5\times10 = 1500 + 250 = 1750$

$= 切2 = m[1,2]+m[3,5] + 5\times10\times10 = 1750 + 500 = 2250$

$= 切3 = m[1,3]+m[4,5] + 5\times5\times10 = 1375 + 250 = 1625$

$= 切4 = m[1,4]+m[5,5] + 5\times20\times10 = 875 + 1000 = 1875$

8.

(1) 若 $x_i = y_i$

則將 $c[i-1, j-1]$ 的值加1寫入 $c[i,j]$ 中(就是左上那格)

左上箭頭可以幫助在最後尋找答案

例:若 x_2 和 y_3 的字母都等於"B"

則將 $c[1,2]$ 的值加1後寫在 $c[2,3]$

再將左上箭頭寫在 $b[2,3]$

(↖)

(2)

i \ j		0	1	2	3	4	5
	y_j		B	D	C	B	A
0	x_i	0	0	0	0	0	0
1	D	0	↑ 0	↖ 1	← 1	← 1	← 1
2	A	0	↑ 0	↑ 1	↑ 1	↑ 1	↖ 2
3	C	0	↑ 0	↑ 1	↖ 2	← 2	↑ 2
4	B	0	↖ 1	↑ 1	↑ 2	↖ 3	← 3
5	D	0	↑ 1	↖ 2	↑ 2	↑ 3	↑ 3
6	A	0	↑ 1	↑ 2	↑ 2	↑ 3	↖ 4

最佳解:DCBA