Algorithm 小考(一)

詳解與配分

Question 1 - (1)

- 詳解

```
n ← length[p] − 1
for i ← 1 to n do
    m[i, i] ← 0
end for
for ℓ ← 2 to n 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
            end if
        end for
    end for
end for
```

$$m[i,j] = \begin{cases} 0 & \text{if } i = j \\ \min_{i \le k < j}\{m[i,k] + m[k+1,j] \\ \quad + p_{i-1} \cdot p_k \cdot p_j\} & \text{if } i < j \end{cases}$$

We have three nested loops:

1. ℓ, length, $O(n)$ iterations
2. i, start, $O(n)$ iterations
3. k, split point, $O(n)$ iterations

Body of loops: constant complexity.

Total complexity: $O(n^3)$

- 配分(10%)
  - Algorithm(or Pseudocode) 12分
  - Time complexity 3分

Question 1 – (2)

- 詳解

M[i][j]

	1	2	3	4	5	6
1	0	15750	7875	9375	11875	15125
2		0	2625	4375	7125	10500
3			0	750	2500	5375
4				0	1000	3500
5					0	5000
6						0

S[i][j]

	2	3	4	5	6
1	1	1	3	3	3
2		2	3	3	3
3			3	3	3
4				4	5
5					5

- So, ANS= $((A_1(A_2A_3))((A_4A_5)A_6))$ minimum number of scalar multiplications = 15125

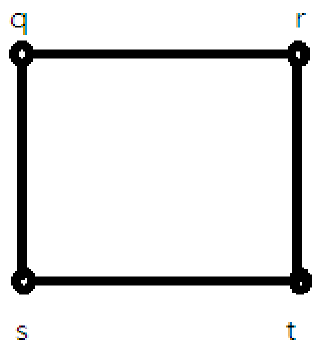
Question 2

- 詳解
  - Overlapping subproblem
    - A recursive algorithm revisits the same subproblem over and over again.
  - Optimal substructure
    - An optimal solution to the problem contains optimal solutions to subproblems.

- 配分(10%)
  - 答案錯全錯 扣十分
  - 只寫出答案，未描述 扣四分

Question 3

- **詳解**
  - Unweighted longest simple path problem does not satisfy the optimal substructure.



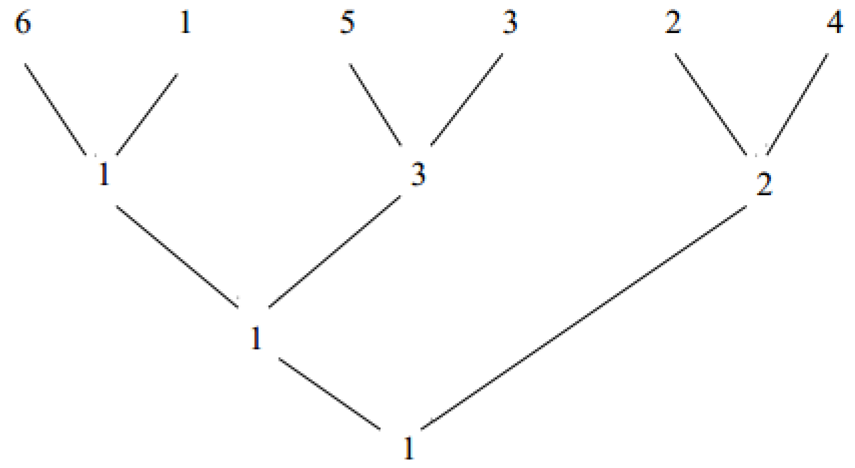
- 從圖可得知 q到t的最長path是q-r-t
  q-r並不是q到r的最長path (應該是q-s-t-r)
  而r-t也不是 r到t的最長path (應該是r-q-s-t)
  所以不具有optimal substructure

- 配分(15%)
  - 答案錯全錯
  - 例子敘述不完整，扣部分分數

Question 4

解答:

Solution: The smallest element is found in $n-1$ comparisons through a cup tournament where the least of two compared elements advance to the next round. After $\lceil \log n \rceil$ rounds the smallest element remains. The second smallest is any of the $\lceil \log n \rceil$ elements that has been compared to the smallest. By $\lceil \log n \rceil - 1$ further comparisons we get the answer. Example:



With 6 elements we get 3 candidates to be the second smallest. The total number of comparisons is $(6-1)+(3-1)=7$.

*註: 3 candidates are 6, 3, 2

so $(n - 1) + (\lceil \log n \rceil - 1) = n + \lceil \log n \rceil - 2$

Question 4

- 解答:

利用winner tree的概念，兩兩元素比較，較小的element當root，以此類推可得整個tree的root為最小值，n個元素總共比了n-1次，找最小花n-1時間，而第二小元素一定有與最小元素比較過

因此把這些element當候選，總共有⌈ log n ⌉個(樹高)，取最小的元素只要⌈ log n ⌉-1次比較，所以找第二小花⌈ log n ⌉-1時間，總共花n-1+⌈ log n ⌉-1 = n+⌈ log n ⌉-2 時間

- 配分(10%)
  - 說明找最小(5%)以及找第二小(5%)的時間複雜度

Question 5

- 解答:

If n is even, compare the first two elements and assign the larger to max and the smaller to min. Then process the rest of the elements in pairs.

If n is even, we do 1 initial comparison and then 3(n – 2) / 2 more comparisons.

$$\text{\# of comparisons } = \frac{3(n-2)}{2} + 1 = \frac{3n-6}{2} + 1 = \frac{3n}{2} - 3 + 1 = \frac{3n}{2} - 2.$$

If n is odd, set both min and max to the first element. Then process the rest of the elements in pairs.

If n is odd, we do 3(n – 1) / 2 = 3⌊ n /2 ⌋ comparisons.

- 配分(15%)
  - 說明找最大(8%)以及找最小(7%)的比較次數

Question 6

解答:

Algorithm:

Step 1: 將元素分成個數為5的組，共⌈n / 5⌉組

Step 2: 找出各組之中位數

Step 3: 利用Recursive求得各組中位數之中位數

Step 4: 令Step 3找的數為X，用PARTITION將n個數分為< n與≥ n兩組

Step 5: If i = k，return X

If i < k，return < n的partition,遞迴呼叫SELECT

If i > k，return ≥ n的partition,遞迴呼叫SELECT

Time complexity:

Steps 1, 2 and 4 each take O (n) time

Step 3 takes time T (⌈ n / 5 ⌉)

Step 5 takes time ≤ T (7n / 10 + 6)

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

解遞迴得T (n) = O (n),