2015 Algorithm Midterm Solutions

指導教授：謝孫源 教授

助教: 葉承翰、詹博丞、林帝丞

Question 1(10pts)

Solution:

- .a.overlapping subproblem
- optimal substructure
- b. without overlapping

Question 2-a(5pts)

Solution:

- 詳解
  - 因為$\frac{P_1}{W_1} = \frac{4}{3}$，$\frac{P_2}{W_2} = \frac{6}{5}$，$\frac{P_3}{W_3} = 1$
  - 取物順序 item1 -> item2 -> item3
  - 先拿item1：因為$W_1 = 6$ 可全拿
  - 再拿item2：因為$W_2 = 5$ 只能拿 2 kg
  - 總共獲利為：$8 + 6 \times \frac{2}{5} = 10.4$
- 配分(5%)
  - 答案錯全錯 扣五分

Question 2-b(5pts)

Solution:

- 詳解

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	8	8	8
0	0	0	0	0	6	8	8	8
0	0	0	3	3	6	8	8	9

  - 所以最大獲利為9
- 配分(5%)
  - 答案錯全錯 扣五分

Question 3(10pts)

Solution:

  - 不行
  - 因為若W[i]為實數，則有可能使得 **k-W[i]**不為整數
  - 因此不可以WORK
- 配分(10%)
  - 沒有解釋 扣六分
  - 解釋錯誤 扣十分

Question 4(10pts)

Solution:

$$e[i, j] = \begin{cases} 0 & \text{if } j = i - 1 , \\ \min\limits_{i \le r \le j} \{e[i, r-1] + e[r+1, j] + w(i, j)\} & \text{if } i \le j . \end{cases}$$

```
OPTIMAL-BST(p, q, n)
for i ← 1 to n + 1
    do e[i, i − 1] ← 0
       w[i, i − 1] ← 0
for l ← 1 to n
    do for i ← 1 to n − l + 1
           do j ← i + l − 1
              e[i, j] ← ∞
              w[i, j] ← w[i, j − 1] + p_j
              for r ← i to j
                  do t ← e[i, r − 1] + e[r + 1, j] + w[i, j]
                     if t < e[i, j]
                       then e[i, j] ← t
                            root[i, j] ← r
return e and root
```

Question 4(10pts)

Solution:

e

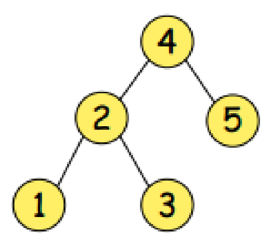
i \ j	0	1	2	3	4	5
1	0	0.15	0.5	0.8	1.35	2.1
2		0	0.2	0.5	0.95	1.65
3			0	0.15	0.5	1.1
4				0	0.2	0.7
5					0	0.3
6						0

w

i \ j	0	1	2	3	4	5
1	0	0.15	0.35	0.5	0.7	1
2		0	0.2	0.35	0.55	0.85
3			0	0.15	0.35	0.65
4				0	0.2	0.5
5					0	0.3
6						0

root

i \ j	1	2	3	4	5
1	1	2	2	2	4
2		2	2	3	4
3			3	4	4
4				4	5
5					5



Cost = 2.1

Question 4(10pts)

Solution:

$$e[1,2]=\min\begin{cases} e[1,0]+e[2,2]+w(1,2)=0.55 & ,r=1 \\ e[1,1]+e[3,2]+w(1,2)=\underline{0.5} & \underline{,r=2} \end{cases}$$

$$e[2,3]=\min\begin{cases} e[2,1]+e[3,3]+w(2,3)=\underline{0.5} & \underline{,r=2} \\ e[2,2]+e[4,3]+w(2,3)=0.55 & ,r=3 \end{cases}$$

$$e[3,4]=\min\begin{cases} e[3,2]+e[4,4]+w(3,4)=0.55 & ,r=3 \\ e[3,3]+e[5,4]+w(3,4)=\underline{0.5} & \underline{,r=4} \end{cases}$$

$$e[4,5]=\min\begin{cases} e[4,3]+e[5,5]+w(4,5)=0.8 & ,r=4 \\ e[4,4]+e[6,5]+w(4,5)=\underline{0.7} & \underline{,r=5} \end{cases}$$

$$e[1,3]=\min\begin{cases} e[1,0]+e[2,3]+w(1,3)=1 & ,r=1 \\ e[1,1]+e[3,3]+w(1,3)=\underline{0.8} & \underline{,r=2} \\ e[1,2]+e[4,3]+w(1,3)=1 & ,r=3 \end{cases}$$

$$e[2,4]=\min\begin{cases} e[2,1]+e[3,4]+w(2,4)=1.05 & ,r=2 \\ e[2,2]+e[4,4]+w(2,4)=\underline{0.95} & \underline{,r=3} \\ e[2,3]+e[5,4]+w(2,4)=1.05 & ,r=4 \end{cases}$$

$$e[3,5]=\min\begin{cases} e[3,2]+e[4,5]+w(3,5)=1.35 & ,r=3 \\ e[3,3]+e[5,5]+w(3,5)=\underline{1.1} & \underline{,r=4} \\ e[3,4]+e[6,5]+w(3,5)=1.15 & ,r=5 \end{cases}$$

Question 4(10pts)

Solution:

$$e[1,4] = \min\begin{cases} e[1,0]+e[2,4]+w(1,4)=1.65 & ,r=1 \\ e[1,1]+e[3,4]+w(1,4)=1.35 & ,r=2 \\ e[1,2]+e[4,4]+w(1,4)=1.4 & ,r=3 \\ e[1,3]+e[5,4]+w(1,4)=1.5 & ,r=4 \end{cases}$$

$$e[2,5] = \min\begin{cases} e[2,1]+e[3,5]+w(2,5)=1.95 & ,r=2 \\ e[2,2]+e[4,5]+w(2,5)=1.75 & ,r=3 \\ e[2,3]+e[5,5]+w(2,5)=1.65 & ,r=4 \\ e[2,4]+e[6,5]+w(2,5)=1.8 & ,r=5 \end{cases}$$

$$e[1,5] = \min\begin{cases} e[1,0]+e[2,5]+w(1,5)=2.65 & ,r=1 \\ e[1,1]+e[3,5]+w(1,5)=2.25 & ,r=2 \\ e[1,2]+e[4,5]+w(1,5)=2.2 & ,r=3 \\ e[1,3]+e[5,5]+w(1,5)=2.1 & ,r=4 \\ e[1,4]+e[6,5]+w(1,5)=2.35 & ,r=5 \end{cases}$$

Question 5(10pts)

- **解答:**
- Define c[i,w] to be the value of the solution for items 1~i and maximum weight w
- $$\begin{cases} \mathbf{0} & \mathbf{if\ i = 0\ or\ w = 0} \\ \mathbf{c[i-1, w]} & \mathbf{if\ w_i > w} \\ \mathbf{max(v_i + c[i-1, w-w_i], c[i-1, w]} & \mathbf{if\ i > 0\ and\ w \geq w_i} \end{cases}$$

Dynamic-0-1-Knapsack(v,w,n,W)

Let c[0..n,0..W] be a new array

For w = 0 to W

  c[0,w] = 0

For i = 1 to n   -O(n)

  c[i,0] = 0

  for w=1 to W   -O(w)

    if $w_i \leq w$

      if v_i + c[i-1,w- w_i]>c[i-1,w]

        c[i,w] = $v_i + c[i-1, w-w_i]$

      else c[i,w] = c[i-1,w]

    else c[i,w] = c[i-1,w]

Question 5(10pts)

- Time complexity : O(nw)
- 配分方式(10%)

  Code: 7 points

  Time complexity analyze: 3 points

Question 6(10pts)

解答:

- 詳解
  - LCS長度 = 6

    <1,0,0,1,1,0> or <1,0,1,1,0,1> or <1,0,1,0,1,1>
  - 若兩序列擺相反則答案為 <0,1,0,1,0,1>
- 配分(10%)
  - 箭號沒畫、共同字沒有圈、只寫長度為6沒有寫答案扣部分分數
  - 有畫圖但整個圖畫錯，扣6~8分

Question 7(5pts)

解答:

- 0-1 knapsack cannot be solved using the greedy strategy
- 配分(5%)
- 答錯全錯 無部份給分

Question 7(5pts)

解答:

item	value	weight
1	8	5
2	4	4
3	5	4

Capacity = 8

- 詳解
  - Greedy: $\frac{8}{5} > \frac{5}{4} > \frac{4}{4}$
    - Choose item1 , <u>Value = 8</u>
  - If choose item2 & item3 , <u>Value = 9</u>
- 配分(10%)
  - 範例不符合　斟酌給分
  - 未給範例　扣十分

Question 8(10pts)

解答:

w[i][j]

	1	2	3	4	5	6
1	0	0.25	0.4	0.6	0.95	1
2		0	0.15	0.35	0.7	0.75
3			0	0.2	0.55	0.6
4				0	0.35	0.4
5					0	0.05
6						0

e[i][j]

	1	2	3	4	5	6
1	0	0.25	0.55	1.05	1.85	2
2		0	0.15	0.5	1.2	1.3
3			0	0.2	0.75	0.85
4				0	0.35	0.45
5					0	0.05
6						0

Question 8(10pts)

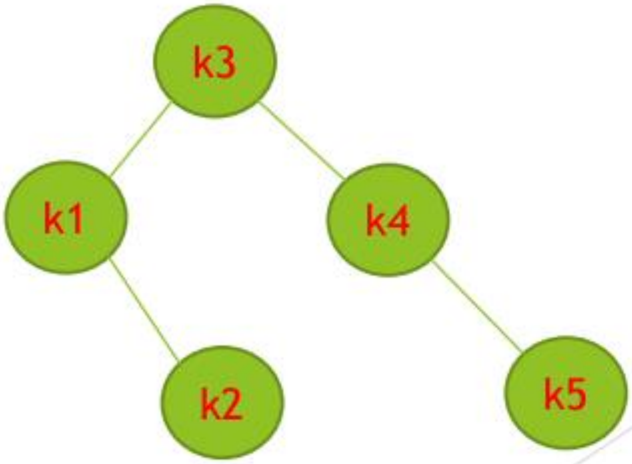
解答:

Root[i][j]

	1	2	3	4	5
1	1	1	2	3	3
2		2	3	3 or 4	4
3			3	4	4
4				4	4
5					5

(a) Cost = 2

(b) Structure of an optimal binary search tree :



*注意是" binary search tree": 小的在左，大的在右

Question 8(10pts)

解答:

- 配分(10%)
- 扣分方式
  - 表格部份算錯1個扣2分，最多扣6分
  - optimal binary search tree畫錯扣2分

Question 9(10pts)

解答:

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

```
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
```

Question 9(10pts)

解答:

m

i \ j	1	2	3	4
1	0	30	66	102
2		0	60	80
3			0	48
4				0

Cost = 102

s

i \ j	2	3	4
1	1	2	2
2		2	2
3			3

$(A_1A_2)(A_3A_4)$

$$m[1,2]=m[1,1]+m[2,2]+p_0p_1p_2=30 \quad ,k=1$$
$$m[2,3]=m[2,2]+m[3,3]+p_1p_2p_3=60 \quad ,k=2$$
$$m[3,4]=m[3,3]+m[4,4]+p_2p_3p_4=48 \quad ,k=3$$
$$m[1,3]=\min\begin{cases} m[1,1]+m[2,3]+p_0p_1p_3=150 & ,k=1 \\ m[1,2]+m[3,3]+p_0p_2p_3=\underline{66 \quad ,k=2} \end{cases}$$
$$m[2,4]=\min\begin{cases} m[2,2]+m[3,4]+p_1p_2p_4=\underline{88 \quad ,k=2} \\ m[2,2]+m[4,4]+p_1p_3p_4=180 & ,k=3 \end{cases}$$
$$m[1,4]=\min\begin{cases} m[1,1]+m[2,4]+p_0p_1p_4=148 & ,k=1 \\ m[1,2]+m[3,4]+p_0p_2p_4=\underline{102 \quad ,k=2} \\ m[1,3]+m[4,4]+p_0p_3p_4=138 & ,k=2 \end{cases}$$

Question 10(10pts)

解答:

```
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
```

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

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) 8分
- Time complexity 2分