Design and Analysis of Algorithms

Middle Exam_solution

1.(10%，錯一格扣一分)

	Insertion sort	Selection sort	Merge sort	Heap sort	Quick sort
Average case	O(n^2)	O(n^2)	O(nlgn)	O(nlgn)	O(nlgn)
Worst case	O(n^2)	O(n^2)	O(nlgn)	O(nlgn)	O(n^2)
Stable or not	Stable	Stable	Stable	Not stable	Not stable

2.(10%, 無使用定義證明得 0 分；只有定義其他過程錯誤得 2 分；有使用定義但過程有瑕疵 or 解釋不清得 7 分)

This is true iff 0 $\leq cn^k \leq \sum_{i=0}^{d} a_i n^i$ for all n >n_0 and some c.

Divide with n^k and obtain 0≤ c $\leq \sum_{i=0}^{d} a_i n^{i-k}$.

Since k $\leq$ d， d-k≥ 0, so n^{d-k} *will be* larger than 1

So,choose c= a_d ⇨ P(n) $\geq cn^k = a_d n^k$.

3.(7%，無使用變數變換全錯)

Let m = lg n, thus,T (2^m) = 3T ($2^{m/2}$) + m, (2 分)

We now rename S(m) = T (2^m) to produce new recurrence

S(m) = 3S(m/2) + m (2 分) => S(m) = Θ($m^{\lg 3}$). (2 分)

Changing back to T (n) and resubstituting m = $\lg n$, T (n) = Θ($(\lg n)^{\lg 3}$) (1 分)

4. (8%，無使用 substitution method 全錯

假設 lgn 且過程正確最高可得 4 分，過程部分錯誤酌扣 1 分)

假設 nlgn 且過程正確最高可得 3 分，過程部分錯誤酌扣 1 分)

We guess $T(n) \leq cn-d$ (2 分)

$T(n) \leq (c\lfloor n/2\rfloor-d)+(c\lceil n/2\rceil-d)+1$ (過程 2 分)

$= cn-2d+1$

$\leq cn-d$ (2 分)

As long as $d \geq 1$, we can choose the constant c let $T(n) = O(n)$. (2 分)

5. (7%，過程有瑕疵 or 不完整扣 3 分)

把每一個 sorted array 的最小值(k 個)拿出來做 min-heap，然後 extract-min，拿掉那一個 array 的元素，就再把那個 array 目前的最小值放入這個 min-heap，如此重複直到所有 array 的元素都排好。

每次拿一個元素做 insert 跟 extract-min 時都需要 O(lgk)的時間，有 n 個元素，所以是 O(nlgk)

簡單的證明，n 個元素裡面的最小值一定是某一個 sorted array 的最小值，所以把所有 array 的最小值拿來做 min-heap 時，extract-min 的就會是 n 個元素中最小的，當把最小的值拿掉時，整個問題除了少了一個元素之外，其他都一樣，當我們把拿掉最小值 array 剩餘的最小值再放進 heap 後，又可以確保剩餘的最小值在這個 heap 裡，如此就可以找到第二小的，重覆這樣的步驟，就能夠排好所有元素

6. (10%，只有分析特定 case 最多得 5 分；過程有瑕疵 or 不完整扣 3 分)

1. Let $a_1, a_2, \ldots, a_n$ denote the set of n numbers initially placed in the array
2. Further, we assume $a_1 < a_2 < \ldots < a_n$ (So, a_1 may not be the element in A[1] originally)
3. Let X_{ij} = # comparisons between a_i and a_j in all Partition calls
4. Then, X = # comparisons in all Partition calls =

   $X_{12} + X_{13}+\ldots+X_{n-1,n}$

   ➔ Average # comparisons

   $= E[X]$

   $= E[X_{12} + X_{13} + \ldots + X_{n-1,n}]$

   $= E[X_{12}] + E[X_{13}] + \ldots + E[X_{n-1,n}]$
5. Pr(ai compared with aj once) = 2/(j-i+1)

   Pr(ai not compared with aj) = (j-i-1)/(j-i+1)

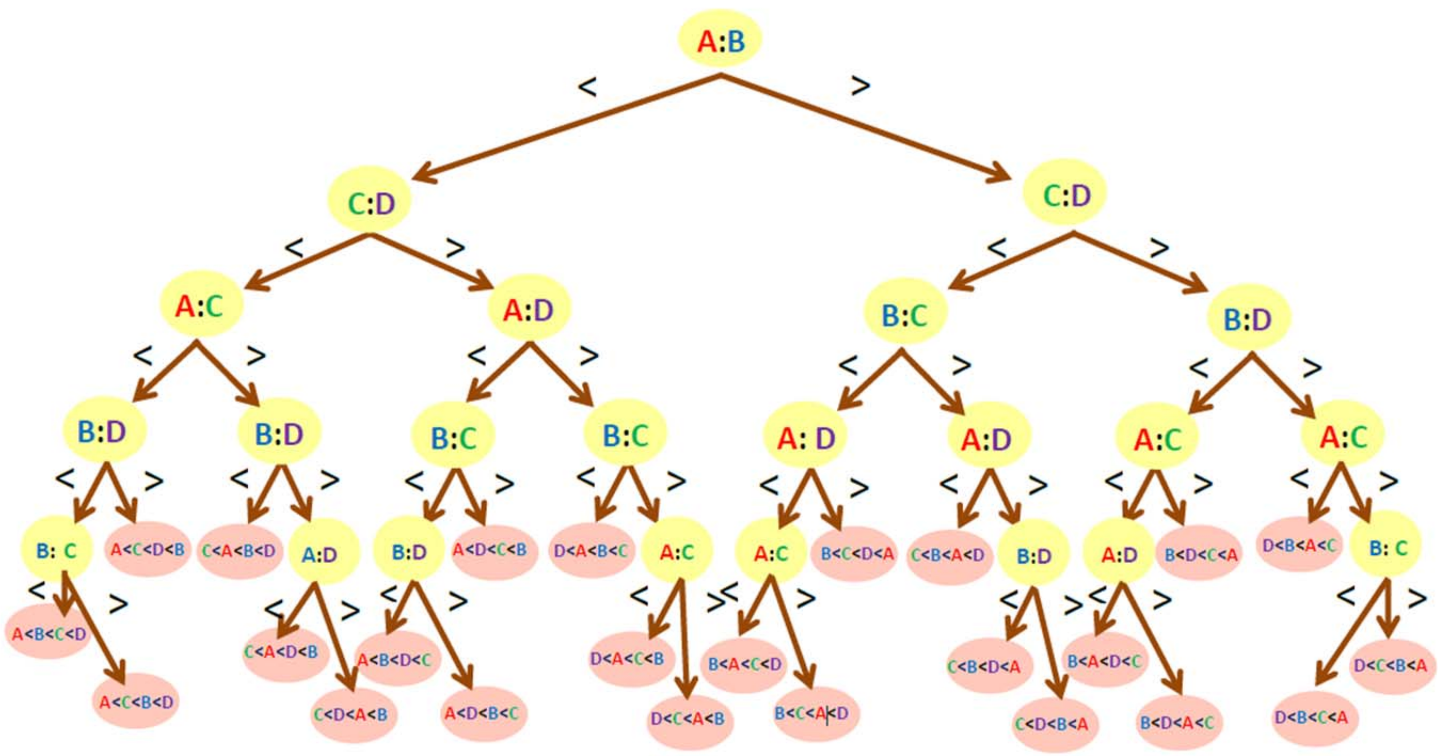
   ➔ $E[X_{ij}] = \; 1 * 2/(j-i+1) \; + \; 0 * (j-i-1)/(j-i+1) = 2/(j-i+1)$
6. $E[X] = S_{i=1 \text{ to } n-1} S_{j=i+1 \text{ to } n}\, 2/(j-i+1)$

   $= S_{i=1 \text{ to } n-1} S_{k=1 \text{ to } n-i}\, 2/(k+1)$

   $< S_{i=1 \text{ to } n-1} S_{k=1 \text{ to } n}\, 2/k$

   $= S_{i=1 \text{ to } n-1} O(\log n) = O(n \log n)$

7. (8%，依照寫對的比例給分 ，非 optimal 不給分)



8. (5% ,有寫出 k=O(n)全對 ,只有寫出限制 k 大小部份給 3 分 ，都沒提及不給分)

No, there is no contradiction with the fact that the lower bound for sorting is O(nlgn), because the linear sorting algorithm assume some properties on the input, and determine the sorted order by distribution.

The running time of the linear sorting algorithm is O(n + k).

Ps: k is the limitation of the input value

➔ if k = O(n), time is (asymptotically) optimal

9.

(10%，分三部分(第一點、二三四、五到九)，各占分(3,4,3)
如果同學使用 CH9 page15 的解法，因為 worst case 的 time complexity 為 $O(n^2)$所以只給三分)

Select(S, k)

/* First, find a good pivot */

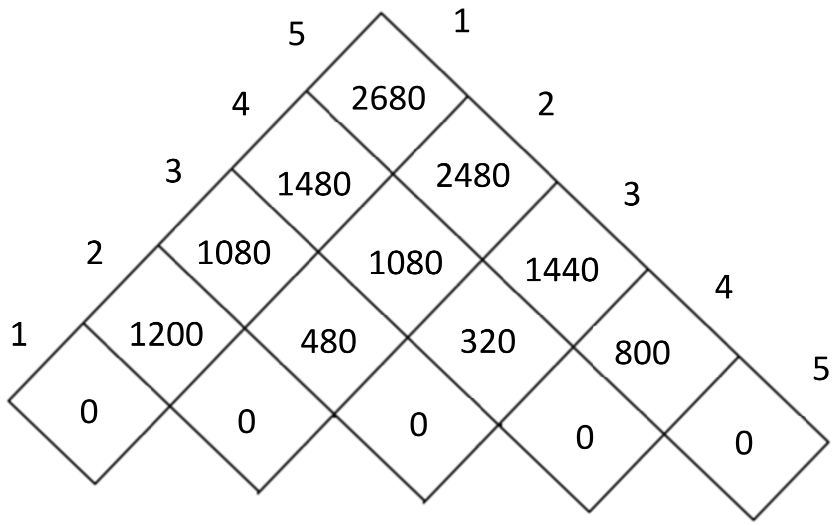
1. If |S| less than a small number then use insertion sort to return the answer
   Else Partition S into $\lceil |S|/5 \rceil$ groups, each group has five items (one group may have fewer items);
2. Sort each group separately;
3. Collect median of each group into S';
4. Find median m of S':
   m = Select(S', $\lceil\lceil |S|/5 \rceil/2\rceil$);
5. Let q = # items of S smaller than m;
6. If (k == q + 1)
   return m;
   /* Partition with pivot */
7. Else partition S into X and Y
   X = {items smaller than m}
   Y = {items larger than m}
   /* Next, form a sub-problem */
8. If (k < q + 1)
   return Select(X, k)
9. Else
   return Select(Y, k–(q+1));

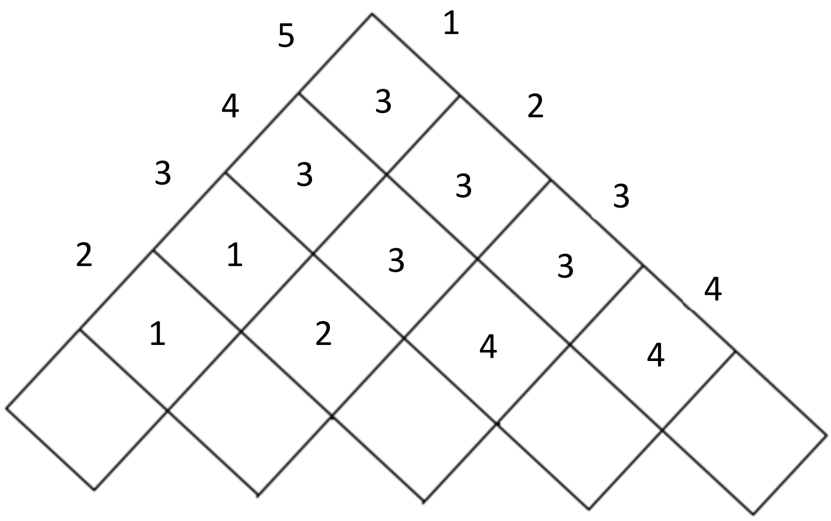
10. (5%，少一項扣 3 分)

- Optimal substructure: an optimal solution to the problem contains optimal solutions to subproblems
- Overlapping subproblems: a recursive algorithm revisits the same subproblem over and over again

11.

(10%，分三部分(方法、計算、解答)，各占分(4,3,3))





$$\boldsymbol{m_{1,3} = \min\{m_{1,2} + m_{3,3} + 320\,,\ m_{1,1} + m_{2,3} + 600\}}$$

$$\boldsymbol{m_{2,4} = \min\{m_{2,3} + m_{4,4} + 600\,,\ m_{2,2} + m_{3,4} + 1200\}}$$

$$\boldsymbol{m_{3,5} = \min\{m_{3,4} + m_{5,5} + 1600\,,\ m_{3,3} + m_{4,5} + 640\}}$$

$$\boldsymbol{m_{1,4} = \min\{m_{1,3} + m_{4,4} + 400\,,\ m_{1,2} + m_{3,4} + 800\,,\ m_{1,1} + m_{2,4} + 1500\}}$$

$$\boldsymbol{m_{2,5} = \min\{m_{2,2} + m_{3,5} + 2400\,,\ m_{2,3} + m_{4,5} + 1200\,,\ m_{2,4} + m_{5,5} + 3000\}}$$

$$\boldsymbol{m_{1,5} = \min\{m_{1,1} + m_{2,5} + 3000\,,\ m_{1,2} + m_{3,5} + 1600\,,\ m_{1,3} + m_{4,5} + 800\,,\ m_{1,4} + m_{5,5} + 2000\}}$$

=>$\boldsymbol{(A_1(A_2A_3))\ (A_4A_5)}$

12. (10%，分三部分(數字、箭頭、答案)，各占分(6,2,2))

		a	b	c	b	d	a	a
	0	0	0	0	0	0	0	0
d	0	0	0	0	0	1	1	1
c	0	0	0	1	1	1	1	1
b	0	0	1	1	2	2	2	2
a	0	1	1	1	2	2	3	3
d	0	1	1	1	2	3	3	3
b	0	1	2	2	2	3	3	3
c	0	1	2	3	3	3	3	3
a	0	1	2	3	3	3	4	4

		a	b	c	b	d	a	a
	0	0	0	0	0	0	0	0
d	0	0	0	0	0	1↖	1←	1←
c	0	0	0	1↖	1←	1←	1←	1←
b	0	0	1↖	1←	2↖	2←	2←	2←
a	0	1↖	1←	1←	2↑	2←	3↖	3↖
d	0	1↑	1←	1←	2↑	3↖	3←	3←
b	0	1↑	2↖	2←	2↖	3↑	3←	3←
c	0	1↑	2↑	3↖	3←	3←	3←	3←
a	0	1↖	2↑	3↑	3←	3←	4↖	4↖

⇨ abca (答案不只一種)