Algorithm
2013 Fall 期中考

解答與配分

Question 1

- 解答

$f(n) = \theta(g(n))$ if and only if 存在三正數c_1, c_2和n_o, 使得$c_1 \times g(n) \leq f(n) \leq c_2 \times g(n)$, for all $n \geq n_o$.

$f(n) = o(g(n))$ if and only if 對任何正數c，會存在一個正數n_o, 使得$f(n) < c \times g(n)$, for all $n \geq n_o$.

$f(n) = \omega(g(n))$ if and only if對任何正數c，會存在一個正數n_o, 使得$f(n) > c \times g(n)$, for all $n \geq n_o$.

- 配分(10%)
- 扣分方式

θ 4分 o ω 3分

正數c -1分

$n \geq n_o$ -1分

$<$, $\leq$ -1分

Question 2

- 解答

利用recursive tree method

$$T(n) = 2T\left(\left\lfloor \sqrt{n} \right\rfloor\right) + \lg n = \lg n + \lg n + \ldots = O(\log\log n \cdot \lg n)$$

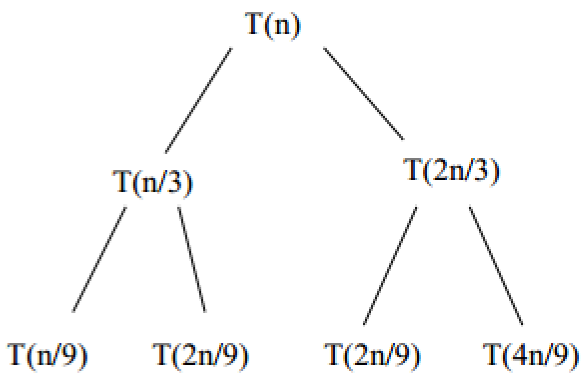
- 配分(10%)

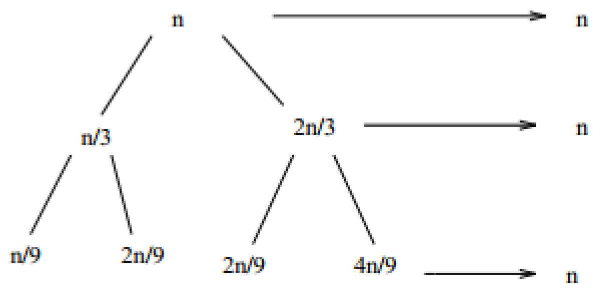
答案錯不給分

Question 3

- 解答

Draw the recursion tree





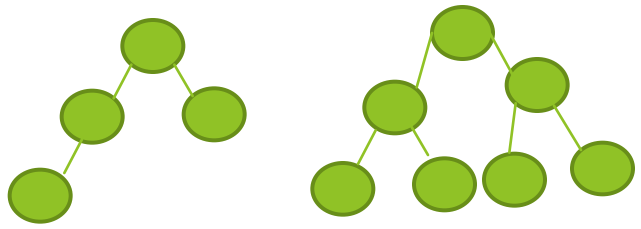
Shortest path to leaf: $n \le 3^k$ or $k \ge \lg_3 n$

longest path to leaf: $n \le (3/2)^k$ or $k \ge \lg_{3/2} n$

題目要我們SHOW出T(n)= Ω(nlgn)所以我們直接找lower bound

T(n) ≥ n log n = Ω(nlgn)

Question 4



- 解答
  - 令樹高為h，高度從1開始，Heap中node個數介於 $2^h \le n \le 2^{h+1}-1$ (第h+1層最少node數~最多node數)
  - 各邊取log後 => $\log 2^h \le \log n \le \log(2^{h+1}-1) < \log 2^{h+1}$

    => $h \le \log n < h+1$

    => h= $\lfloor logn \rfloor$
- 配分(10%)
- 扣分方式
  - 最多node和最少node case都要考慮進去，只寫一半扣3~5分

Question 5

- 解答
  - Not a max heap beacause A[9]=7 is bigger than its parents A[4]=6
- 配分(10%)
- 扣分方式
  - 答案錯全錯

Question 6

- 解答
  - Heap sort: Build a heap out the first element from all k streams. Remove the min and insert the next element into the heap from the stream to which min belongs. Keep repeating this till we run out of all streams
  - Merge sort: Arrange these k lists by pairwise merging. It needs O(n) to merge k lists. Merging needs lgk times so the height is lgk. So the answer is O(n*lgk)
- 配分(10%)
- 扣分方式
  - 使用heap sort 或merge sort皆可
  - 需簡單說明此algo為何是O(n*lgk) time，只寫出algo扣3~5分
  - 只畫簡圖沒有簡單說明扣3~5分

Question 7

- 解答

```
QUICKSORT(A, p, r)
if p < r
  then q ← PARTITION(A, p, r)
       QUICKSORT(A, p, q − 1)
       QUICKSORT(A, q + 1, r)

Initial call is QUICKSORT(A, 1, n).
```

```
PARTITION(A, p, r)
x ← A[r]
i ← p − 1
for j ← p to r − 1
    do if A[j] ≤ x
          then i ← i + 1
               exchange A[i] ↔ A[j]
exchange A[i + 1] ↔ A[r]
return i + 1
```

- 配分(10%)
- 扣分方式

有寫即得2分 依完整度斟酌扣分 QuickSort(佔2分) Partiton(佔6分)

Question 8

- 解答

寫出linear sort algo.的作法，再針對題目給的條件進行策略:

1) Radix Sort

策略重點: 為了 達到linear time的時間複雜度，勢必要以越大的基底k(空間換取時間)，可自己假設基底k大小，已知d = $\log_k$(n^3)，時間複雜度為O(d*(n+k))，為了讓d視為一個常數，可假設k=n，則時間複雜度為O(3*(n+n))，即為O(n)。

1) Counting Sort

策略重點:已知key range k為n^3，故時間複雜度為O(n+k)，為了讓k達到linear time，可以將每個元素實施[key_i%n]作為排序依據，則key range 為0~n-1，代表只需O(n)，此題為n^3，可做3個回合:Pass1- [key_i%n]實施Counting Sort後將結果帶入Pass2- [(key_i/n)%n]、 Pass2做完結果再帶入Pass3- [(key_i/(n^2))%n]做Counting Sort ，即可得結果，故時間複雜度為O(n+O(n)+O(n)+O(n))，即為O(n)。

Question 8

- 配分(10%)
- 扣分方式

寫出linear sort algo.的作法即得4分

未說明出針對題目給的n個integers 和 range:0~n^3 - 1的策略 扣4~6分

Question 9

- 解答
  - $(n-1)+\lceil \log n \rceil-1 \; = \; (n-2) + \lceil \log n \rceil$
- 配分(10%)
- 扣分方式
  - 未寫出高斯符號扣兩分
  - 只寫答案無過程扣五分
  - 證明過程不完整斟酌扣分

Question 10

- 解答
  - Consider a decision tree of height h with l reachable leaves corresponding to a comparison sort on n elements. Because each of the n! permutations of the input appears as some leaf, we have n! ≤ l . Since a binary tree of height h has no more than 2^h leaves, we have $n! \leq l \leq 2^h$.
  - By taking logarithms , Implies $h \geq \log(n!) = \Omega(n\log n)$
- 配分(10%)
- 扣分方式
  - 證明過程小錯誤扣兩~三分
  - 證明不完整扣五分
  - 未使用決策樹(Decision Tree)證明全錯