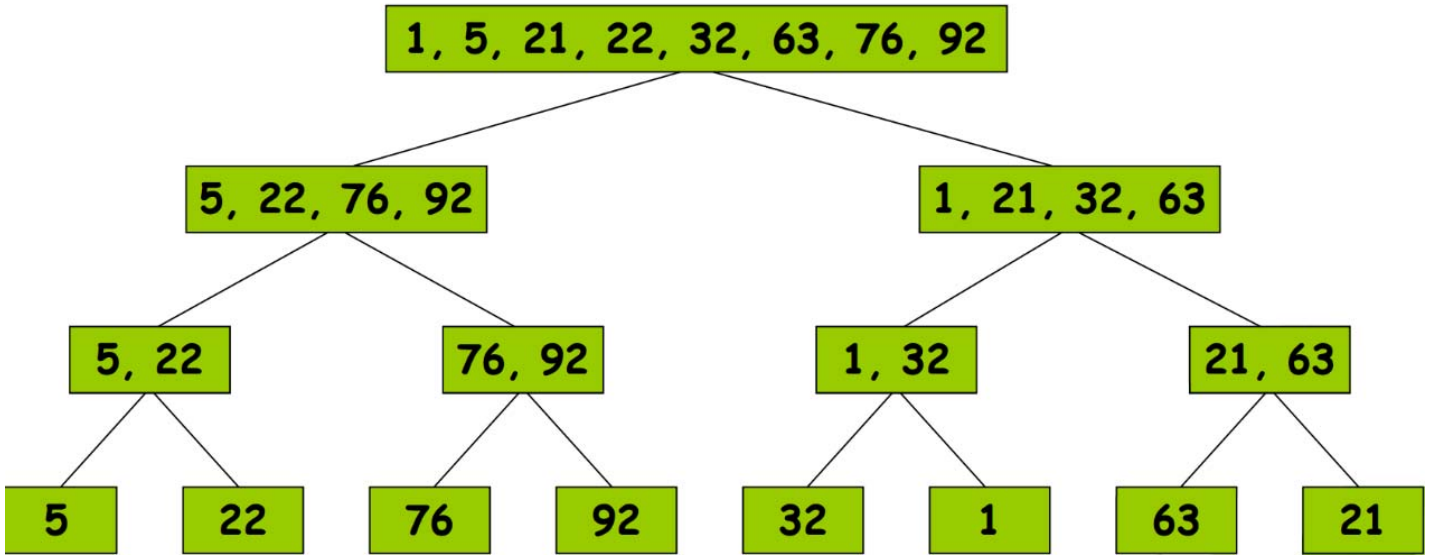
2016 Algorithm HW1 Solutions

指導教授：謝孫源 教授

助教: 盧緯 詹博丞 楊順翔 許景添

Question 1(10pts)

Solution:



Question 2(10pts)

Solution:

i. By Master Method Case 3: $T(n) = \Theta(n^3)$

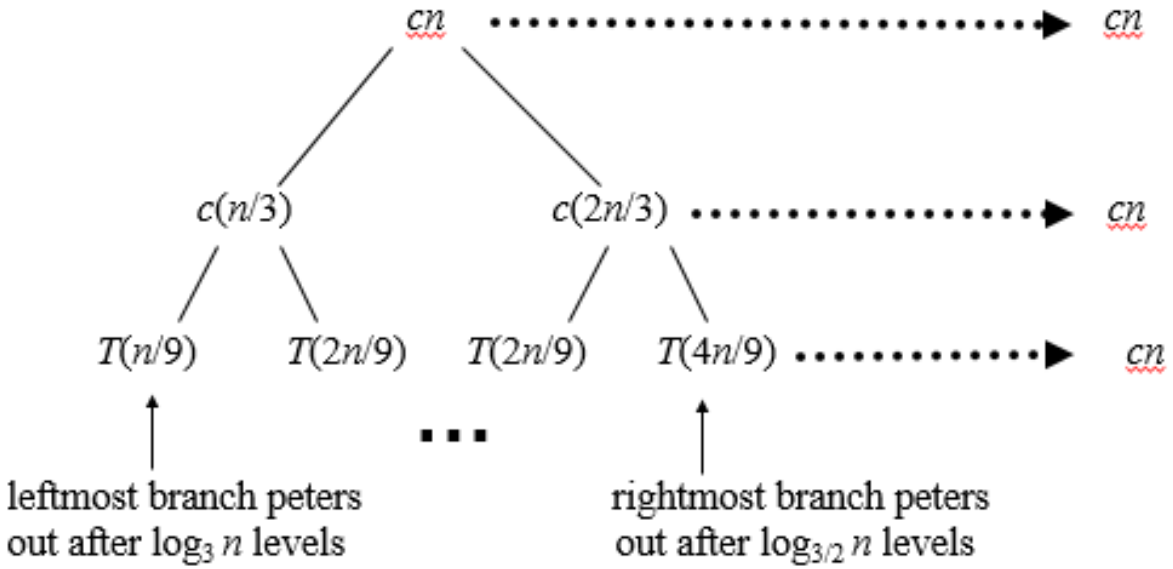
ii. By Master Method Case 3: $T(n) = \Theta(n)$

iii. By Substitution method: $T(n) = T(n) = \Theta(n^2)$

Question 3(10pts)

Solution:

Use to generate a guess. Then verify by substitution method.
E.g.: $T(n) = T(n/3) + T(2n/3) + \ominus(n)$. For upper bound, rewrite as $T(n) \leq T(n/3) + T(2n/3) + cn$; for lower bound, as $T(n) \geq T(n/3) + T(2n/3) + cn$. By summing across each level, the recursion tree shows the cost at each level of recursion (minus the costs of recursive calls, which appear in subtrees):



Question 4(10pts)

Solution:

With a=4,b=2, we have f(n)=$n^2 \lg n \neq O(n^{2-\epsilon}) \neq \Omega(n^{2-\epsilon})$, so no - we cannot apply the master method.

Let's guess $\Theta(n^2 \lg^2 n)$:

$$T(n) \leq 4T\left(\frac{n}{2}\right) + n^2 \lg n$$

$$\leq 4c\left(\frac{n}{2}\right)^2 \lg^2\left(\frac{n}{2}\right) + n^2 \lg n$$

$$\leq cn^2 \lg\left(\frac{n}{2}\right) \lg n - cn^2 \lg\left(\frac{n}{2}\right) \lg 2 + n^2 \lg n$$

$$\leq cn^2 \lg^2 n - cn^2 \lg n \lg 2 - cn^2 \lg\left(\frac{n}{2}\right) + n^2 \lg n$$

$$\leq cn^2 \lg^2 n + (1-c)n^2 \lg n - cn^2 \lg\left(\frac{n}{2}\right) \qquad (c > 1)$$

$$\leq cn^2 \lg^2 n - cn^2 \lg\left(\frac{n}{2}\right)$$

$$\leq cn^2 \lg^2 n$$

Question 5(10pts)

Solution:

- $\log a^{\log_b c} = \log_b c \times \log_e a = \log_b c \times \frac{\log_b a}{\log_b e} = \frac{\log_b a \times \log_b c}{\log_b e}$
- $\log c^{\log_b a} = \log_b a \times \log_e c = \log_b a \times \frac{\log_b c}{\log_b e} = \frac{\log_b a \times \log_b c}{\log_b e}$

Question 6(10pts)

Solution:

- 先算O: (5pts)
  - Log($n!$)=log(n)+log($n-1$)+⋯+log(1)
  - ≤log(n)+log(n)+⋯+log(n)=$n$log$n$
  - 所以log($n!$)=O($n$log$n$)
- 再算Ω: (5pts)
  - Log($n!$)=log(n)+log($n-1$)+⋯+log($\frac{n}{2}$)+⋯+log(1)
  - ≥log($\frac{n}{2}$)+log($\frac{n}{2}$)+⋯+log($\frac{n}{2}$)=($\frac{n}{2}$)log($\frac{n}{2}$)
  - 所以log($n!$)=Ω($n$log$n$)
- 有上述可得log($n!$)=θ($n$log$n$)

Question 7(10pts)

Solution:

The best case happens when the partition is even, that is: $T(n) = 2T(n/2) + \Theta(n)$ Using the master method, we get the solution $\Theta(n \lg n)$.

Question 8(10pts)

Solution:

Show that there are at most $\lceil n/2^{h+1} \rceil$ nodes of height h in any n-element heap.

- 解答:

  特別注意:此處的height為從leaf開始往上算，而depth才是從root開始算，所以leaf的height為0

  證明: By induction on number of nodes at height h.

  Base case : Number of leaves at height h=0=number of leaves in the n-element heap. Parent of the last node is the $\lceil n/2 \rceil$-node. This will be the last parent in the heap. All the nodes after this one will be leaves. Therefore the number of leaves is $\lceil n/2 \rceil$. This is also true for the formula, therefore base case is true.

  We assume that number of nodes at height h-1 is given by the formula, $\lceil n/2^h \rceil$.

  Prove height =h

  Note that if we remove all the leaves from the heap, the nodes that were earlier at height 1 now become leaves in the new heap, i.e. they have height 0. Similarity, all the nodes in the new tree will have height that is one less than their old height. We shall use this to prove the induction.

Question 8(10pts)

Solution:

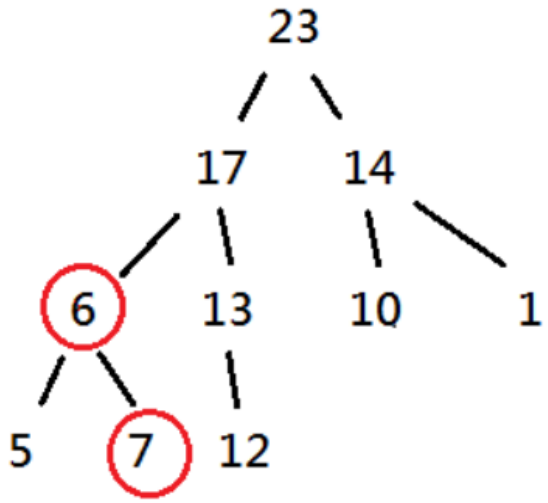
- 解答:

  Induction: Consider a new tree that is obtained by removing the leaves. This tree is a heap with n- $\lceil n/2 \rceil$ nodes in it. The number of nodes at height h in the old heap will be the same at the number of nodes at height h-1 in the new heap, which is $(n-\lceil n/2 \rceil)/2^h = \lceil n/2^{h+1} \rceil$
- 配分(10%)
- 扣分方式
  - 只畫圖舉例未實際證明只給2分

Question 9(10pts)

Solution:

No, max-heap child值須小於parent



答案隊給5分，有畫出正確的圖或說明給5分

Question 10(10pts)

Solution:

- By induction on number of nodes at height *h*.
- Base case : Number of leaves at height *h*=0=number of leaves in the -element heap. Parent of the last node is the $\left\lceil \frac{n}{2} \right\rceil$-node. This will be the last parent in the heap. All the nodes after this one will be leaves. Therefore the number of leaves is $\left\lceil \frac{n}{2} \right\rceil$.This is also true for the formula, therefore base case is true. We assume that number of nodes at height h-1 is given by the formula, $\left\lceil \frac{n}{2^h} \right\rceil$.
- Prove height = *h* Note that if we remove all the leaves from the heap, the nodes that were earlier at height 1 now become leaves in the new heap, i.e. they have height 0. Similarity, all the nodes in the new tree will have height that is one less than their old height. We shall use this to prove the induction.
- Induction: Consider a new tree that is obtained by removing the leaves. This tree is a heap with n- $\left\lceil \frac{n}{2} \right\rceil$ nodes in it. The number of nodes at height h in the old heap will be the same at the number of nodes at height h-1 in the new heap, which is $\frac{n-\left\lceil \frac{n}{2} \right\rceil}{2^h} = \left\lceil \frac{n}{2^{h-1}} \right\rceil$