圖論演算法期中考(參考解答)

1. (a)

A rooted tree is a tree in which one of the vertices is distinguished from the others.

(b)

The binomial tree B_k is an ordered tree defined recursively as follows.

- the binomial tree B_0 consists of a single node.
- B_k consists of two B_{k-1} that are linked together: the root of one is the leftmost child of the root of the other.

(c)

A B-tree T is a rooted tree having the following properties:

- Every node x has the following fields:
  - n[x], the number of keys in node x,
  - $key_1[x] \le key_2[x] \le \cdots \le key_n[x]$,
  - leaf[x], leaf[x] = TRUE if x is a leaf, leaf[x] = FALSE if x is an internal node.
- Each internal node x also contains n[x]+1 pointers $C_1[x], C_2[x], \ldots, C_{n[x]+1}[x]$ to its children.
- If k_i is any key stored in the subtree with root $C_i[x]$, then $k_1 \le key_1[x] \le k_2 \le key_2[x] \le \cdots \le key_{n[x]}[x] \le k_{n[x]+1}k1$.
- All leaves have the same depth, which is the tree's height h.
- Every node x other than the root must have $t - 1 \le n[x] \le 2t - 1$, where $t \ge 2$ is the minimum degree of the B-tree.
- If the tree is nonempty, the root has $1 \le n[root] \le 2t - 1$.

(d)

A mergeable heap is any data structure that supports the following five operations, in which each element has a key:

- MAKE-HEAP() creates and returns a new heap containing no elements.
- INSERT(H, x) inserts element x, whose key field has already been filled in, into heap H.
- MINIMUM(H) returns a pointer to the element in heap H whose key is minimum.
- EXTRACT-MIN(H) deletes the element from heap H whose key is minimum, returning a pointer to the element.
- UNION(H_1, H_2) creates and returns a new heap that contains all the elements of heaps $H_1$and H_2. Heaps H_1 and H_2 are "destroyed" by this

operation.

(e)

The (Binary) heap data structure is an array object that can be viewed as a nearly complete binary tree.

- A binary tree with n nodes and depth k is complete iff its nodes correspond to the nodes numbered from 1 to n in the full binary tree of depth k.

(f)

A binomial heap H is a set of binomial trees that satisfies the following binomial-heap properties.

1. Each binomial tree in H is min-heap ordered:
   key(x) ≥ key(p(x)).
2. For any nonnegative integer k, there is at most one binomial tree in H whose root has degree k.'

(g)

Fibonacci heaps support the mergeable-heap operations and the following two operations.

- DECREASE-KEY(H, x, k) assigns to element x the new key value k, which is assumed to be no greater than its current key value.
- DELETE(H, x) deletes node x from heap H.

2. (a)

存在兩正常數 C=1, $n_0 = \frac{1}{2}$, 使得$f(n) \le cn^2$, for all n≥ n_0, 所以 f(n) = $O(n^2)$

(b)

存在兩正常數 C=$\frac{1}{4}$, $n_0 = 2$, 使得$f(n) \ge cn^2$, for all n≥ n_0, 所以 f(n) = $\Omega(n^2)$

(c)

LR: Assume that $f(n) \in \Theta(g(n))$. So there are positive c, d, n_0 s.t.

$$c|g(n)| \le |f(n)| \le d|g(n)| \quad \text{for all } n \ge n_0 .$$

This implies the desired result. That is, we have $f(n) \in \Omega(g(n))$ because

$$c|g(n)| \le |f(n)| \quad \text{for all } n \ge n_0 ,$$

and we have $f(n) \in O(g(n))$ because

$$|f(n)| \le d|g(n)| \quad \text{for all } n \ge n_0 .$$

RL: Assume $f(n) \in \Omega(g(n))$ and $f(n) \in O(g(n))$. So there are positive c, d, n_0', n_0'' s.t.

$$c|g(n)| \le |f(n)| \quad \text{for all } n \ge n_0'$$

and

$$|f(n)| \le d|g(n)| \quad \text{for all } n \ge n_0'' .$$

Take $n_0 = n_0' + n_0''$. It follows that

$$c|g(n)| \le |f(n)| \le d|g(n)| \quad \text{for all } n \ge n_0 .$$

Thus, $f(n) \in \Theta(g(n))$.

(d)

在 $\sum_{1 \le k \le n} O(n) = n \cdot O(n)$ 有錯

$O(1) + O(2) + \cdots + O(n) = O(n) \neq O(n^2)$

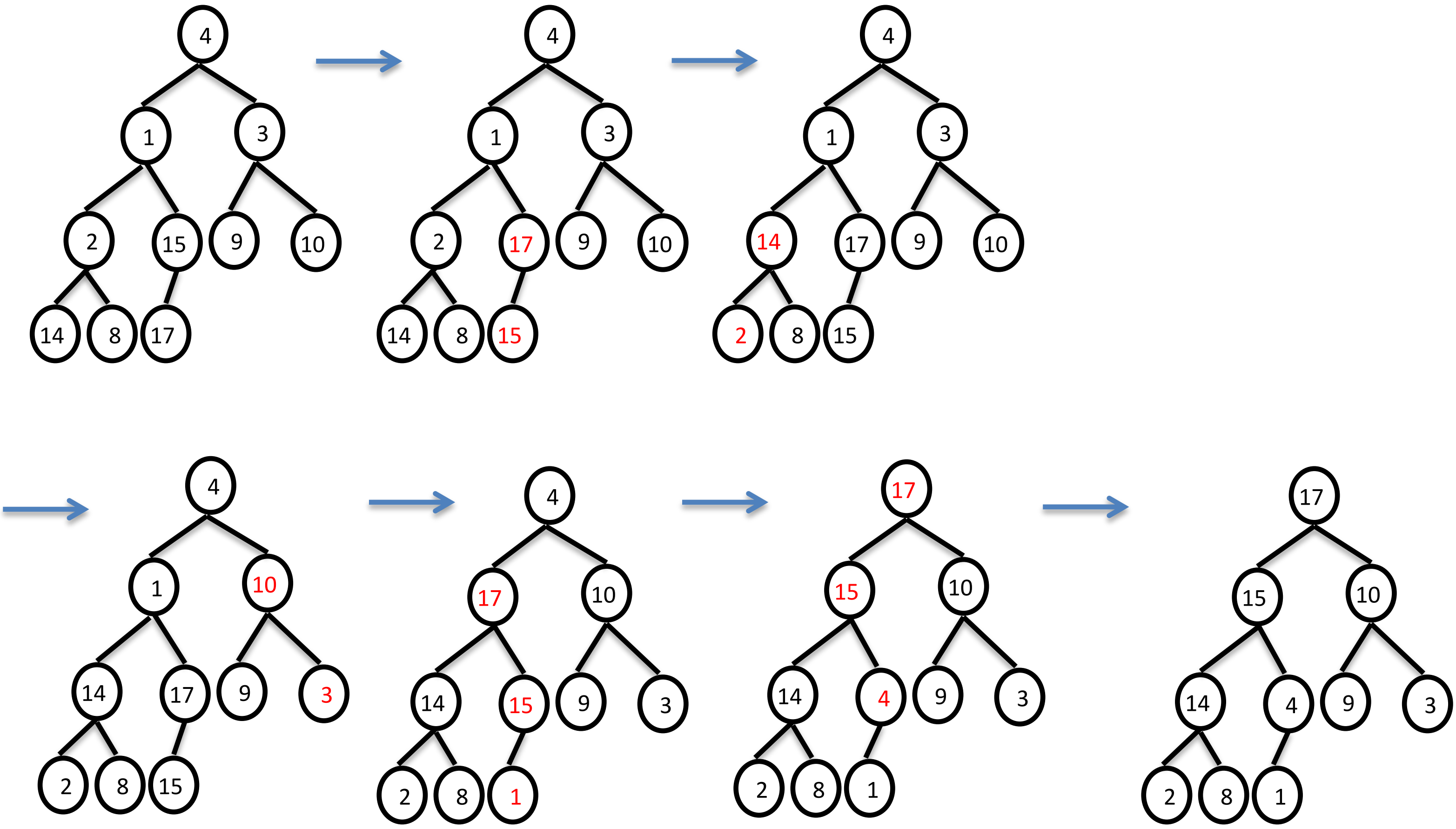
3.

MAX-HEAPIFY procedure takes O(h) time

$$\sum_{h=0}^{\lfloor \lg n \rfloor} \left\lceil \frac{n}{2^{h+1}} \right\rceil * O(h) = O\left(n \sum_{h=0}^{\lfloor \lg n \rfloor} \frac{h}{2^h} \right)$$

$$\sum_{h=0}^{\infty} \frac{h}{2^h} = 2 \left(\text{因爲} \sum_{k=0}^{\infty} k x^k = \frac{x}{(1-x)^2} \right)$$

$$O\left(n\sum_{h=0}^{\lfloor \lg n \rfloor}\frac{h}{2^h}\right)=O\left(n\sum_{h=0}^{\infty}\frac{h}{2^h}\right)=O(n)$$



4.

(a)

Analysis(I):

- Worst-case cost of MULTIPOP is O(n).
- Have n operations.
- Therefore, worst-case cost of sequence is $O(n^2)$.

Analysis(II):

- Each object can be popped only once per time that it’s pushed.
- At most n objects are pushed into S.
- Have $\le$ n PUSHes $\Rightarrow$ $\le$ n POPs, including those in MULTIPOP.
- Therefore, total cost = O(n).
- Average cost of an operation = O(1).

(b)

operation	actual cost	amortized cost
PUSH	1	2
POP	1	0
MULTIPOP	min(k, s)	0

Intuition: When pushing an object, pay 2.

- \$1 pays for the PUSH.
- \$1 is prepayment for it being popped by either POP or MULTIPOP.
- Since each object has \$1, which is credit, the credit ≥ 0.
- Therefore, total amortized cost ≤ 2n, is an upper bound on total actual cost.
- Average cost of an operation = O(1)

(c)

- Φ= # of objects in stack.
- D_0= empty stack ⇒ $\Phi(D_0) = 0$.
- Since # of objects in stack ≥ 0, $\Phi(D_i) \geq 0 = \Phi(D_0)$ for all i.

operation	actual cost	$\Phi(D_0) - \Phi(D_{i-1})$	amortized cost
PUSH	1	(s + 1) − s = 1	1 + 1 = 2
POP	1	(s − 1) − s = −1	1 − 1 = 0
MULTIPOP	k'= min(k, s)	(s − k') − s = −k'	k' − k' = 0

s = # of objects initially.

Therefore, amortized cost of a sequence of n operations= $\sum_{i=1}^{n} \hat{c}_i = O(n)$

5.

Proof:

- The root contains at least one key.
- Thus, there are at least 2 nodes at depth 1.
- All other nodes contain at least t − 1 keys.
- So, at least 2t nodes at depth 2, at least $2t^2$ nodes at depth 3, and so on.

Then, we have $n \geq 1 + (t-1)\sum_{i=1}^{h} 2t^{i-1}$

$$= 1 + 2(t-1)\left(\frac{t^h - 1}{t-1}\right)$$

$$= 2t^h - 1 \qquad h \leq \log_t \frac{n+1}{2}$$

6.

(a)

The height of the tree is k.

- Two copies of B_{k-1} are linked to form B_k.

- Maximum depth in B_k = Maximum depth in $B_{k-1} + 1$.
- By the inductive hypothesis, this maximum depth is (k−1) + 1 = k.

(b)

Let D(k, i) be the number of nodes at depth i of binomial tree B_k.

$$\mathrm{D(k, i)} = D(k-1, i) + D(k-1, i-1)$$
$$= \binom{k-1}{i} + \binom{k-1}{i-1}$$
$$= \binom{k}{i}$$

(c)

- The only node with greater degree in B_k than in B_{k-1} is the root, which has one more child than in B_{k-1}.
- Since the root of B_{k-1} has degree k − 1, the root of B_k has degree k.