Data Structures

Final exam. Spring'03 (Yen)

作答在答案卷上; 題目卷不必繳回

1. (10 pts) Answer the following questions:

1. When discussing binomial heaps we talk about a type of trees called *binomial trees*. Suppose that the root *r* of a binomial tree has 6 children. How many grandchildren does *r* have? (Note: a grandchild of *r* means a child of a child of *r*.) Answer: 15
2. Write down the worst-case running times of the following: (a) *quicksort*, (b) *decrease-key* for *pairing heaps*. Answer: (a) $O(n^2)$, (b) O(1)
3. Which of the choices below best describes the worst-case performance of the Union-Find data structure (with union by rank and path compression) of performing a sequence of *m* operations on *n* elements? Pick the strongest bound which is valid in the worst case. *(A) O(m+n), (B) O((m+n)log n), (C) O((m+n)log log n), (D) O((m+n)log log log n)*. Answer: (D)
4. What is the running time of Dijkstra's algorithm when implemented by Fibonacci heaps. Write your answer in terms of |V| and |E|, i.e., the number of vertices and the number of edges, respectively. Answer: |E|+|V|log|V|
5. Use *recursive depth-first search* to traverse the graph shown in Figure 1 with node 1 as the starting node. Write down the traversal sequence. (At any moment, if more than one node can be visited next, always select the one with the smallest value to visit first.) Answer: 1 4 6 2 3 5 7 8

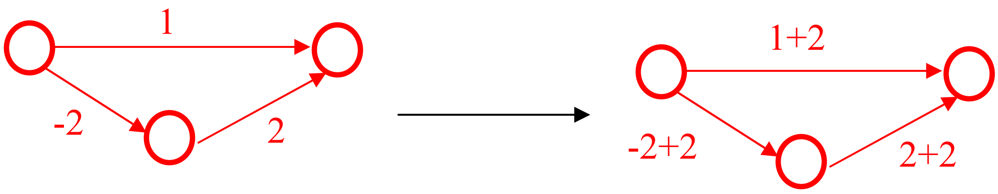
2. (20 pts) *True or False*? (***Score=Max{0, Right – 1/2 Wrong}***; so be careful when making uninformed guesses). No explanation is needed.

1. (F) The average-case running time of INSERTION SORT is O(n).
2. (F) The total amortized cost of a sequence of n operations (i.e., the sum over all operations) gives a lower bound on the total actual cost of the sequence.
3. (F) Let G=(V,E) be a weighted graph and M be a minimum spanning tree of G. The path in M between any pair of vertices v_1 and v_2 must be a shortest path from v_1 to v_2 in G.
4. (F) Let *p* be a shortest path from some source vertex *s* to some other vertex *t* in a graph. If the weight of each edge in the graph is increased by one, *p* remains a shortest path from s to *t*.
5. (F) The *breadth first search* algorithm makes use of a stack.
6. (T) Dijkstra's algorithm is an example of a *greedy* algorithm.
7. (F) Heapsort, quicksort, and mergesort are all asymptotically optimal, stable comparison-based sort algorithms.
8. (T) If each operation on a data structure runs in *O(1)* amortized time, then *n* consecutive operations run in *O(n)* time in the worst case.
9. (F) The height of a leftist heap of *n* nodes is guaranteed to be *O(log n)*.
10. (T) Given a graph whose edge weights are all distinct (i.e., different), then its minimum spanning tree is unique.

3. (10 pts) It is known that Dijkstra's algorithm does not work for graphs with negative edges. Consider the following modification to Dijkstra's algorithm for `solving' the shortest path problem for graphs with negative edges but without negative cycles.

> *If some of the edge weights in a graph are negative, add a large constant C to each edge weight, where C is chosen large enough that every resulting edge weight will be nonnegative, then apply Dijkstra's algorithm to the resulting graph to find shortest paths.*

Does the modified algorithm correctly solve the shortest path problem? Why? Justify your answer. (If your answer is no, find a counterexample; otherwise, explain in a convincing way why the modified algorithm works.) Answer: NO.



4. (8 pts) Given the following facts:

(1) A binary heap can be constructed from an unordered array of numbers in linear (i.e., (O(n)) worst-case time.

(2) Decrease-key and delete-min of a binary heap can be done in O(log n) worst-case time.

(3) Comparison-based sorting requires Ω(n log n).

(4) In-order, Pre-order, and Post-order tree traversals can be done in O(n) time.

(5) Heapsort can be done in O(n log n) worst-case time.

(6) Heapsort is a comparison-based sort algorithm.

Explain how to use SOME of the above statements to show that converting a binary heap into a binary search tree in O(n) worst-case time is *NOT* possible.

Answer:

O(n) O(n) ? O(n)

a seq. of numbers → binary heap → BST → in-order traversal → sorted seq.

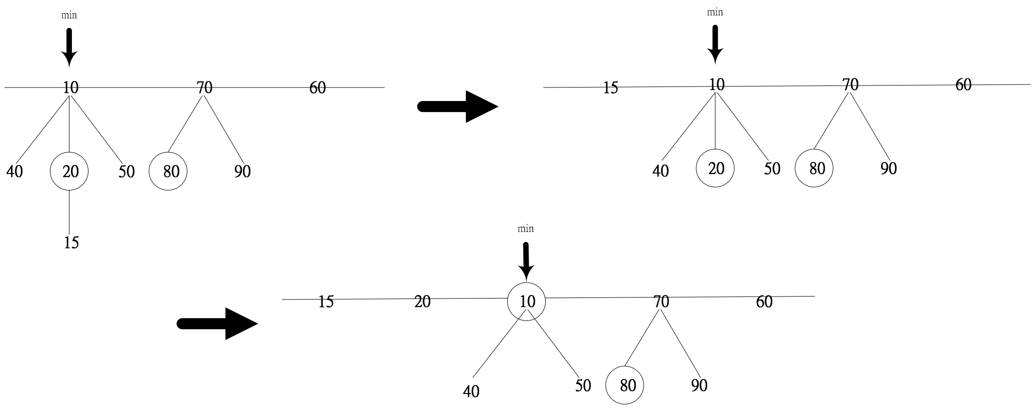
5. (12 pts) Consider the *Fibonacci heap* shown in Figure 2. Nodes that are marked are indicated by dark circles.

(**A:** *8 pts*) Suppose we perform the following two operations in a sequence:

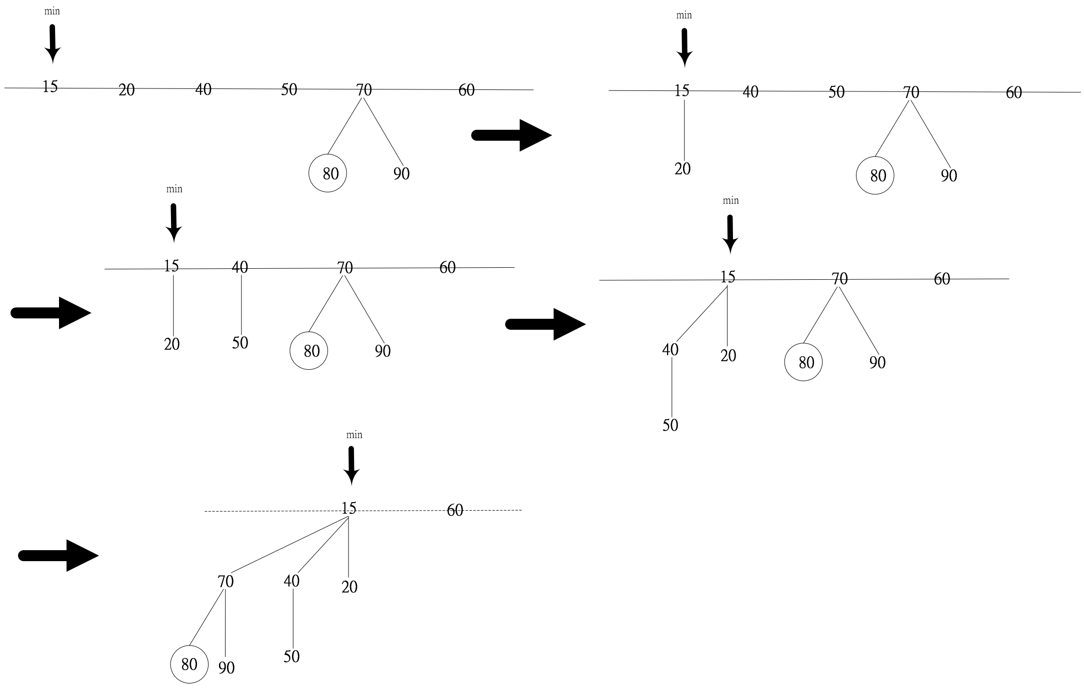
(1) *Decrease-key*: decrease the value 30 to 15, and (2) *Delete-min*. What is the resulting Fibonacci heap? Show your derivation in sufficient detail.

Answer:

Decrease-key 30 -> 15:



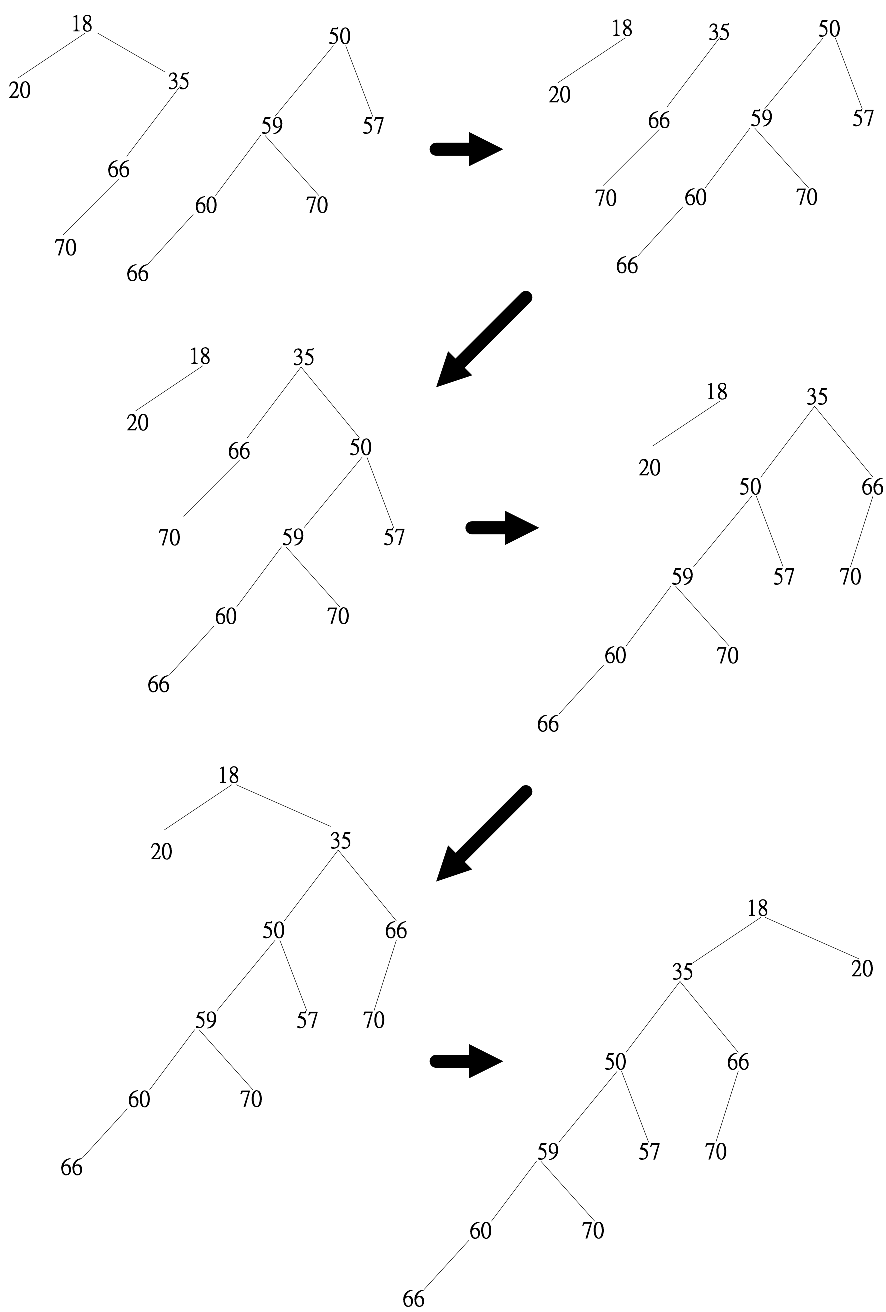
Delete-min



(**B:** *4 pts*) Also state the worst-case running time of the following operations on Fibonacci heaps: *find-min* *O(1)*, *delete-min* *O(n)*, *union* *O(1)*, and *decrease ke y* *O(n)*.

6. (10 pts) Draw the *leftist heap* resulting from performing the *delete-min* operation on the leftist heap shown in Figure 3. Show your derivation in detail.

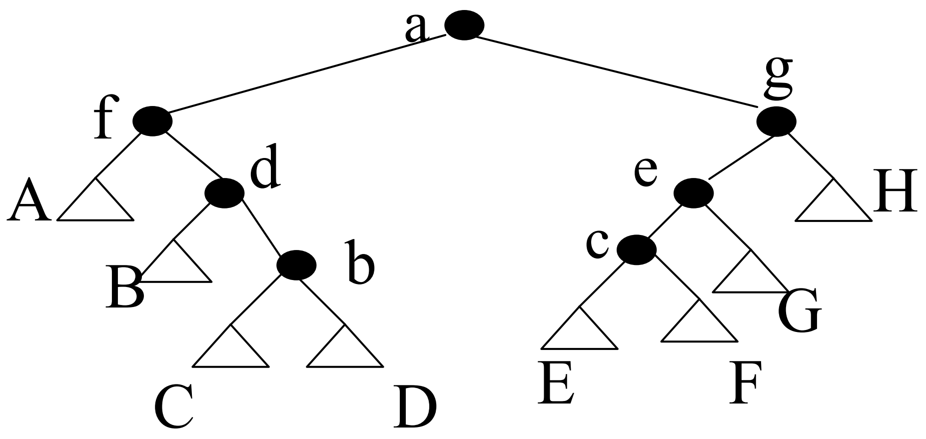
Answer:



7. (10 pts) Suppose we perform a top-down *splay(a)* operation on the splay tree shown in Figure 4, in which *A, B, …, H* are subtrees and *a, b, …, g* are node labels (not key values). Draw the resulting tree. Show your derivation in detail. .

Answer:

(detail skipped)



8. (10 pts) Sort the sequence **7, 28, 5, 18, 123, 1, 14, 45, 89, 4, 78, 2** using *shell sort* with the increments **4** and **1**. Show each of the major intermediate steps.

Answer:

Increment 4 sort:

7, 28, 5, 18, **123**, 1, 14, 45, **89**, 4, 78, 2

7, **28**, 5, 18, 89, **1**, 14, 45, 123, **4**, 78, 2

7, 1, **5**, 18, 89, 4, **14**, 45, 123, 28, **78**, 2

7, 1, 5, **18**, 89, 4, 14, **45**, 123, 28, 78, **2**

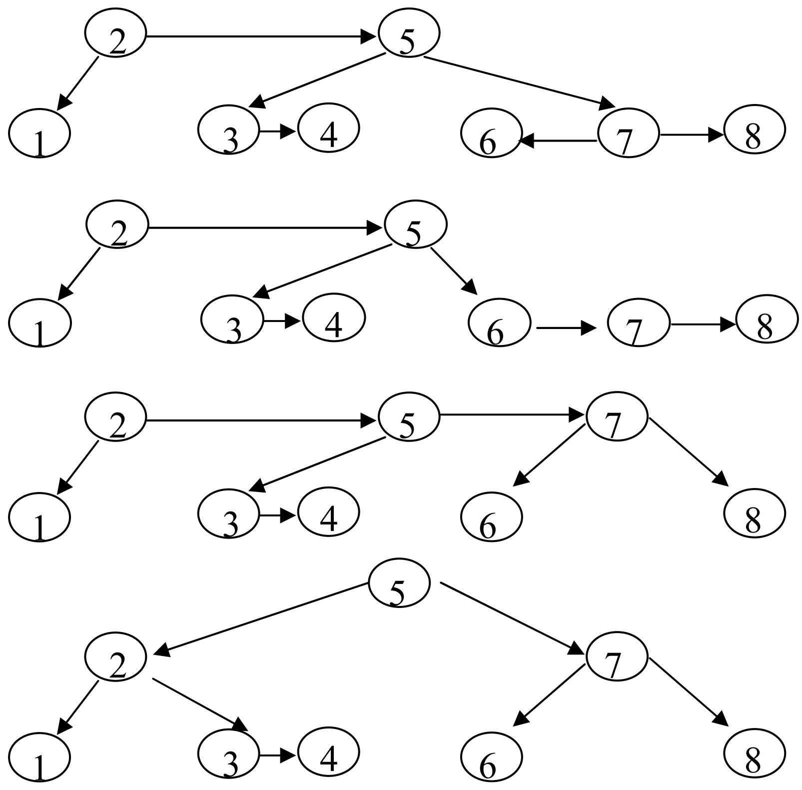
7, 1, 5, 2, 89, 4, 14, 18, 123, 28, 78, 45

Increment 1 sort (usual insertion sort):

1, 2, 4, 5, 7, 14, 18, 28, 45, 78, 89, 123

9. (10 pts) Insert 6 into the *AA-tree* shown in Figure 5. Show your derivation in detail.

Answer:



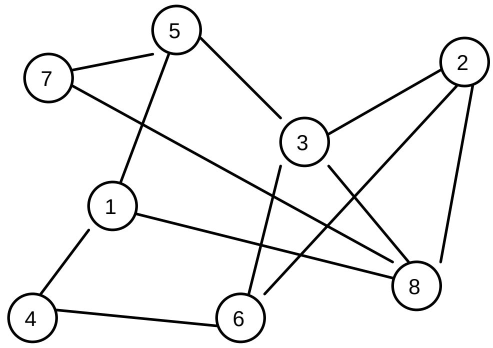


Figure 1

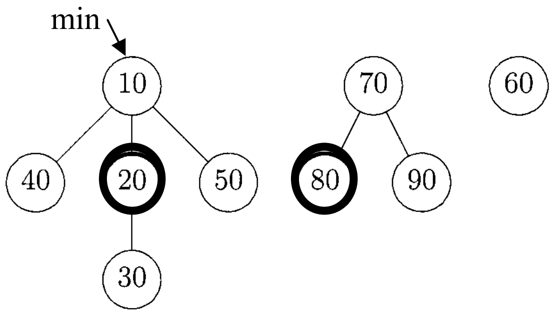


Figure 2

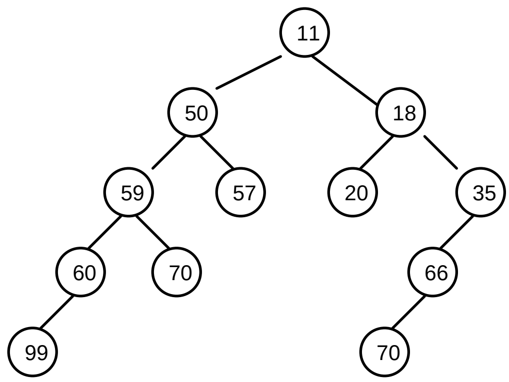


Figure 3

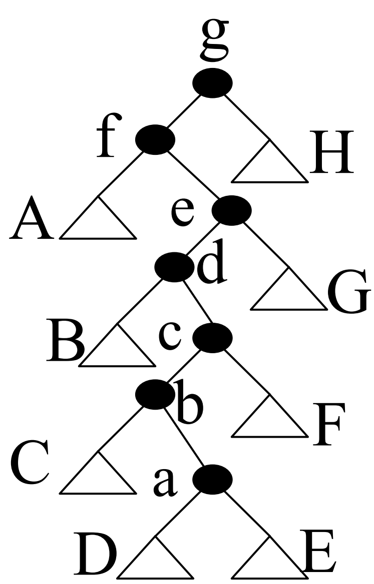


Figure4

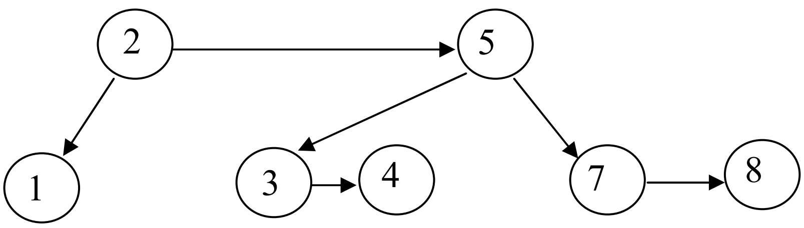


Figure 5

☺ *Have a nice summer!* ☺