UNIVERSITY OF BIRMINGHAM

School of Computer Science

Data Structures & Algorithms

Sample Exam Questions

Data Structures & Algorithms

1. Consider a recorded embedded B+Tree where each leaf block can contain a maximum of 3 records, each internal block can contain a maximum of 3 keys, all record blocks in the tree are fully occupied with 3 records each and the records have key values: $5, 10, 15, \ldots$, and there are 4 record blocks in the file

   (i) Draw this tree in a single diagram **[5 marks]**

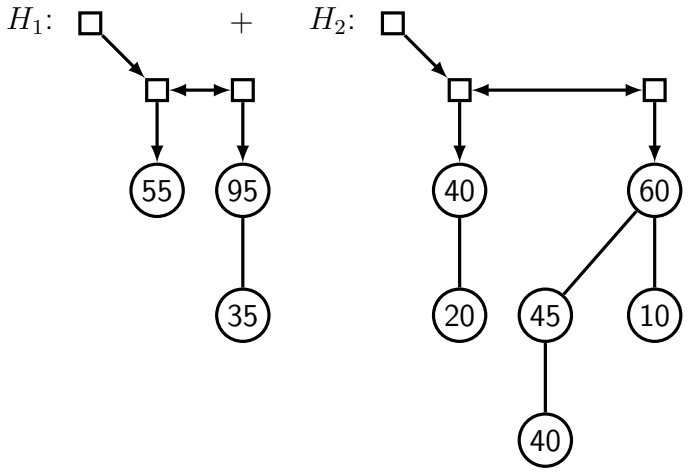
   (ii) Draw the resulting tree after inserting a record with key value 33 into this tree **[10 marks]**

   (iii) State how many disk block writes are necessary to carry out the insertion above and explain exactly which writes were required. **[5 marks]**

   Topics: 06 - B-Trees

2. (i) Consider a Binary Heap where the highest priority is the highest value. The standard Binary Heap `heapify` method is used to construct this Binary Heap on an Array that contains, in index locations 1 to 10, the integers 1 to 10. Choose and write out an ordering of these integers in the array that **maximises** the number of swaps that the `heapify` method will execute on the array and report the total number of swaps executed. **[10 marks]**

   (ii) Draw the result of merging the following two Binomial Heaps:



[10 marks]

   Topics: 07 - Priority Queues and Heap Trees

3. Consider inserting following values into a max priority queue (where the highest priority is the highest numeric value) in the given order:

   10, 50, 25, 80, 35, 20, 65, 40, 30, 75, 15, 45, 90

(i) Assuming a Binary Heap Tree is used to implement the priority queue, draw the resulting binary heap tree both in tree form and in underlying array form using the standard array representation for binary heap trees **[5 marks]**

(ii) Assuming a Binomial Heap is used to implement the priority queue, draw the resulting Binomial Heap **[10 marks]**

(iii) Explain the sequence of steps on both implementations to delete the highest priority element and draw the resulting data structures. **[10 marks]**

Topics: 07 - Priority Queues and Heap Trees

4. Quicksort is being applied to an array containing, in the given order,

   5, 3, 8, 7, 2, 9, 1, 6, 4

   Assume that the pivot selection strategy is to pick the first element in the array, and that the in-place (unstable) partitioning algorithm is used.

   For the first execution of the partition algorithm only, draw the array contents initially and immediately after **each** execution of the call to swap in this algorithm, identifying on your drawing which index location the variables `leftmark` and `rightmark` refer to. **[10 marks]**

   Topics: 08 - Sorting

5. The objects A to G are inserted, in this order, into a hash table. The Hash1 and Hash2 values for these objects are:

	A	B	C	D	E	F	G
Hash1:	17	31	6	14	22	4	5
Hash2:	3	9	5	1	3	7	1

Assume that, in all cases, a maximal load factor of 50% is maintained and that the underlying array has initial capacity of 4 and is doubled in capacity as necessary to maintain the maximal load factor.

(i) Draw a diagram of the resulting hash table if it is implemented with **direct chaining**, where Hash1 is used as the hash value of each object. **[5 marks]**

(ii) Draw a diagram of the resulting hash table if it is implemented with **linear probing**, where Hash1 is used as the hash value of each object. **[5 marks]**

(iii) Draw a diagram of the resulting hash table if it is implemented with **double hashing**, where Hash1 is used as the primary hash value of each object and Hash2 is used as the secondary hash value of each object. **[5 marks]**

Topics: 09 - Hashing and Hash Tables

6. Consider a weighted undirected graph with 5 nodes A, B, C, D, and E, with all weights in the range 1 to 9.

(i) Draw such a graph so that, when the Dijkstra algorithm is applied to it to find the shortest path from A to B, at every iteration of the algorithm a shorter path from A to B is found. **[5 marks]**

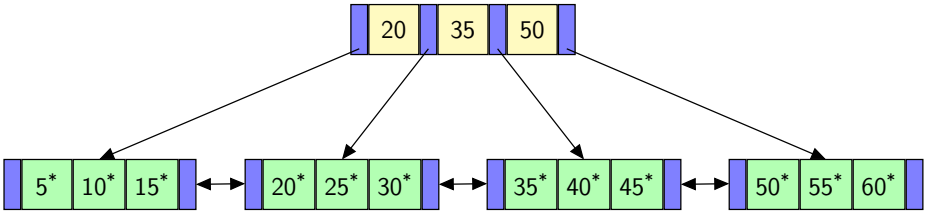
(ii) Demonstrate the execution of the algorithm on this graph by writing out the table of execution steps in the same format as was given in the lectures. **[10 marks]**

Topics: 10 - Graphs

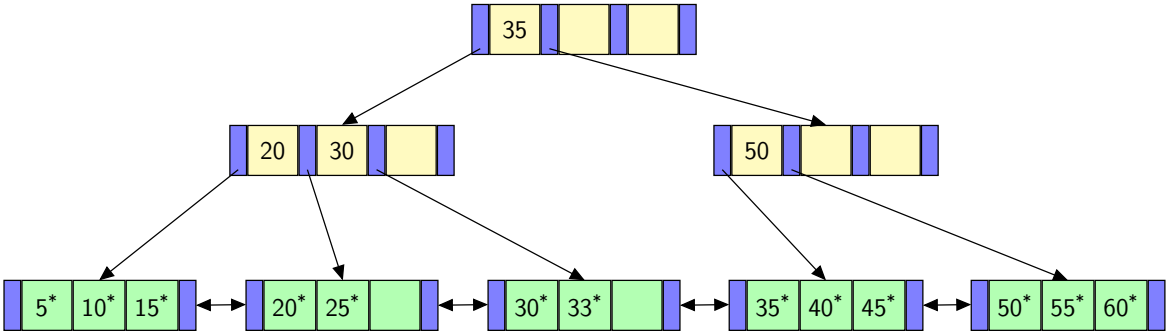
Model answer

1.

(i)



(ii)



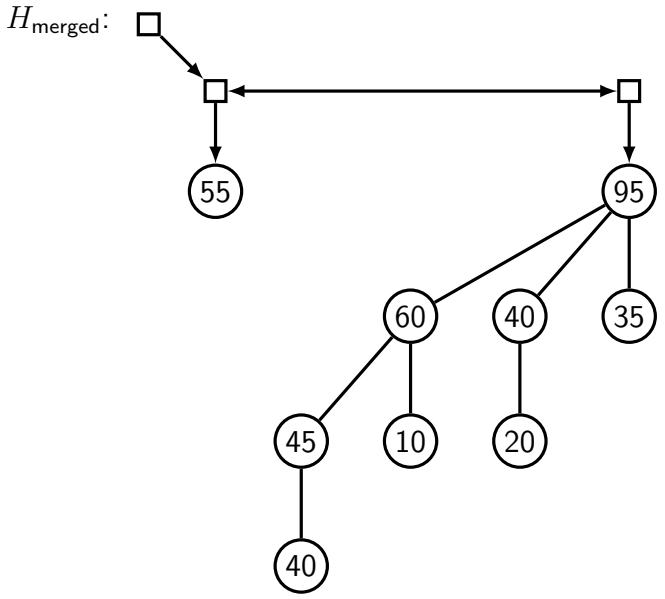
(iii) Every block that has changed must be written, even if it is only a forward or backward pointer that has changed in it. In this case, 6 blocks need to be written:

- The 3 leaf blocks containing $[20^*, 25^*], [30^*, 33^*]$ and, because of the backward pointer, $[35^*, 40^*, 45^*]$
- The 2 internal block at level 1
- The root node block

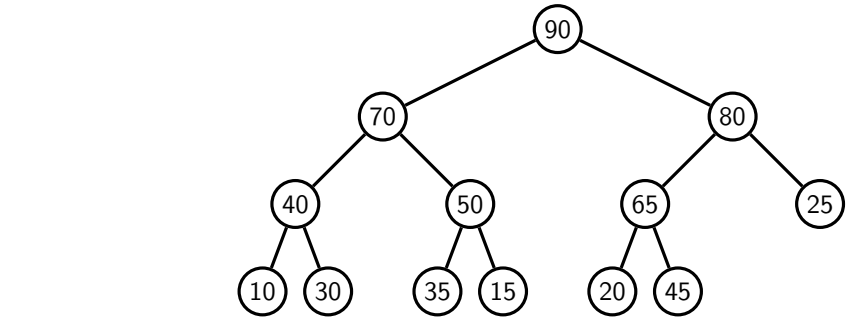
2. (i) There are many different orderings that will ensure that the maximum number of swaps are required, but the simplest is to have them in reverse priority order. Since the highest priority is the highest value, that means reverse priority order is increasing numeric value, i.e. $1, 2, 3, \ldots, 10$.

   With this ordering the heapify algorithm will iterate, starting at the last internal node and continuing up through all the internal nodes until finally arriving at the root node. In each iteration it will trigger a sequence of swaps so that that internal node with be swapped on a full path from its current location down to the leaf level to end up as a leaf node.

   (ii) The result must be exactly as follows to earn full marks

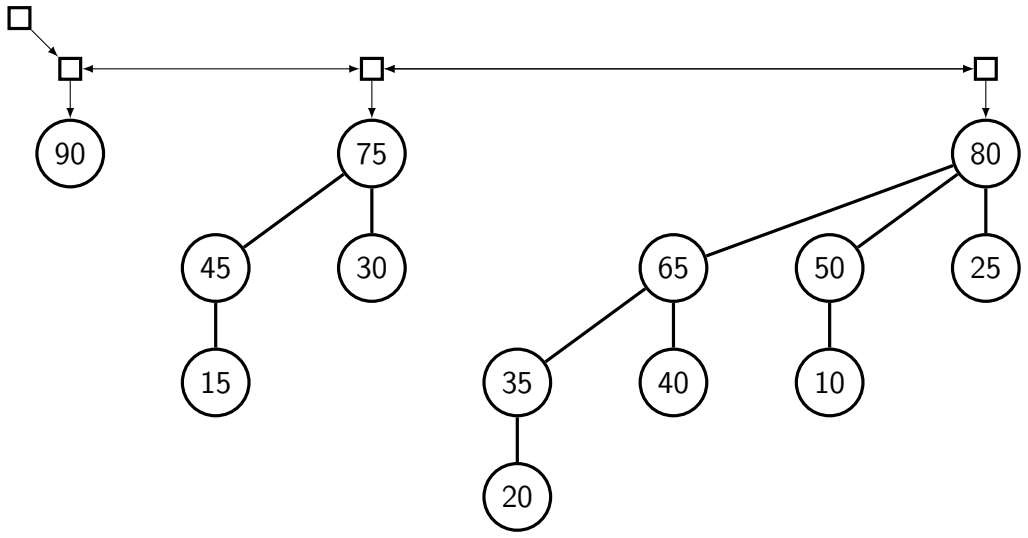


3. (i) This is the ONLY fully correct answer:



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
	90	70	80	40	50	65	25	10	30	35	15	20	45			...

(ii) This is the ONLY fully correct answer:



(iii) **Binary Heap:** The 90 at the root of the tree will be returned. The last element, 45, is removed from the end of the tree and replaces the 90 at the root. This new rott value is bubbled down; first swapping with the 80 (which becomes the new root), then with the 65 (which becomes the right child of the root, and stops there as the left child of the 65 (it does not swap with the 20 which is its left child and it has no right child).

Binomial Heap The linked list of Binomial Trees is searched linearly to find the one with the highest priority root value. In this case it is the first one with root value of 90. This root value is the one to be returned. At this point the usual process is to remove this Binomial Tree from the Binomial Heap, make a new Binomal Tree out of the list of immediate children of the root, and then merge this new Binomial Heap into the original Binomial Heap (minus the Binomial Tree that has been removed. In this case, the 90 has no children so there is no merging that needs to be done.

4. The initial array is:

5	3	8	7	2	9	1	6	4

We only show the state of the array after each swap within one call to partition, without recursing. The resulting states **AFTER each swap** are (leftmark/rightmark identified as L/R respectively):

Swap 5 and 4 to put the pivot out of the way at the end of the array:

L							R	
4	3	8	7	2	9	1	6	5

Swap 8 and 1:

			L	R				
4	3	1	7	2	9	8	6	5

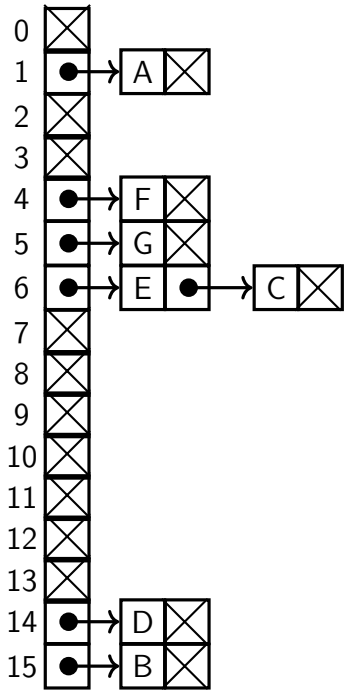
Swap 7 and 2:

			R	L				
4	3	1	2	7	9	8	6	5

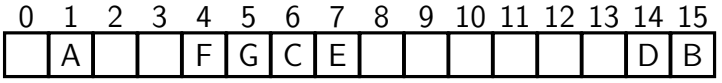
Swap 7 and 5 to put the pivot back between the lower and upper partitions:

4	3	1	2	5	9	8	6	7

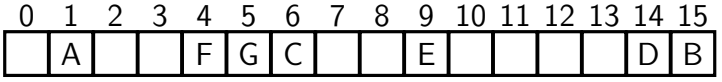
5. (i)


0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
A
F
G
E
C
D
B

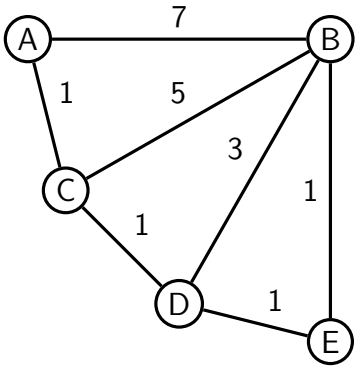
(ii)


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
A F G C E D B

(iii)


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
A F G C E D B

6. (i)



(ii)

A	B	C	D	E	finished
0,A	∞,B	∞,C	∞,D	∞,E	
0,A✓	7,A	1,A	∞,D	∞,E	A
0,A✓	6,C	1,A✓	2,C	∞,E	C
0,A✓	5,B	1,A✓	2,C✓	3,D	D
0,A✓	4,B	1,A✓	2,C✓	3,D✓	E
0,A✓	4,B✓	1,A✓	2,C✓	3,D✓	B