CS 251, LE 2 **Fall 2016**

MIDTERM 2

Tuesday, November 1, 2016

Version 00 - KEY

W1.) (i) Show one possible valid 2-3 tree containing the nine elements: 1 3 4 5 6 8 9 10 12.

Ask your PSO TA if questions

(ii) Draw the final binary search tree after **inserting** the ten integers 57, 85, 35, 9, 47, 20, 26, 99, 93, 10 starting with 57 and ending with 10.

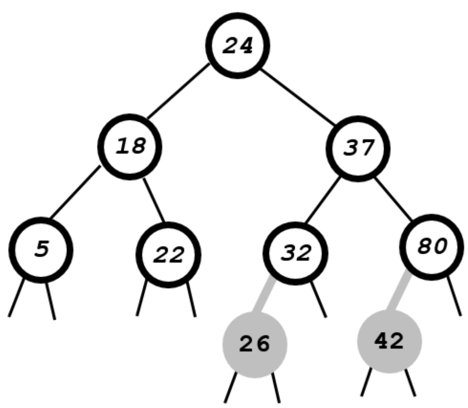
Ask your PSO TA if questions

List the nodes of your tree in **preorder**:

(iii) Consider the LL RB tree shown below. Red nodes and links are shown in **grey**.

Its black height is: __

Show the tree **after** the insertion of key 88. How many rotations are done?



W2.) (i) The 10 keys 39, 0, 1, 19, 12, 3, 10, 46, 20, 5 are inserted into a hash table of **size 9** (in that order) using separate chaining to handle collisions. For x mod 9 used as the hash function, show the hash table and lists after all insertions.

Ask your PSO TA if questions

(ii) The 10 keys 39, 0, 1, 19, 27, 3, 16, 46, 20, 15 are inserted into a hash table of **size 13** (in that order) using linear probing and x mod 13 as the hash function. Show the hash table after all insertions have been done.

Ask your PSO TA if questions

(iii) Draw a Merkle tree having 7 leaves. For leaf 7, mark the path sibling nodes used in a challenge-response.

Ask your PSO TA if questions

1. I have spelled out and bubbled in **correctly** my first name, last name, and Purdue ID.

   A True **B** False

2. The number of your exam version is **00**.
   I have bubbled in the version number on the Scantron sheet under TEST/QUIZ NUMBER.

   A True **B** False

   NOTE

   - Always check the **tightest bound possible**.
   - The height of a tree is **maximum number of edges** on a path from the root to a leaf.

3. Which statement is **not true** for hashing with linear probing?

   A Hashing with linear probing needs to distinguish between free and available cells.

   B To monitor the load, hashing with linear probing needs to maintain the number of entries in the hash table.

   C ** To avoid all collisions, hashing with linear probing sets $\lambda = 0.25$.

   D An entry x hashed to location h(x) may be placed at location (h(x)-10) mod N.

4. The search time in a hash table of size N containing n item has what worst-case performance for separate chaining:

   A $O(N)$

   B $O(N + n)$

   C ** $O(n)$

   D $O(1)$

   E $O(n \log n)$

5. Which statements are **true**?
   1. For linear probing, performance degrades quickly for $\lambda > 0.5$.
   2. For separate chaining, the average cost of an unsuccessful search is $1 + \lambda$.
   3. Double hashing always outperforms linear probing.
   4. $h(x) = x$ **mod** 1023 is generally a good hash function as 1023 is a prime number.

   A All

   B ** 1 and 2

   C 1, 2, and 3

   D 2 and 4

   E 3 and 4

6. A good hash function should be deterministic, i.e., equal keys produce the same hash value.

   A ** True **B** False

7. A Binary Search Tree (BST) stores keys in the range 37 to 573. Suppose the BST has been unsuccessfully searched for **key 273**. Which of the sequences given below list nodes in an order we **could have** encountered them in the search?

   1. 81, 537, 102, 439, 285, 376, 305
   2. 52, 97, 121, 195, 242, 381, 472
   3. 142, 248, 520, 386, 345, 270, 307
   4. 550, 149, 507, 395, 463, 402, 270

   A 1

   B 2

   C ** 3

   D 4

   E None

8. Using no additional entries (only the standard search-entries), the minimum key in a search tree on n nodes can be found **in O(1) time** in what type of search trees?

   A Binary search tree

   B 2-3 tree

   C Left-Leaning Red-Black tree

   D ** None of the above

9. T is a binary search tree. The successor of node u in T is always a node in the subtree rooted at u.

   A True **B** ** False

10. Alice built a binary search tree of height $\lceil \log n \rceil$. Bob built a binary search tree of height $\lfloor \log(4n) \rfloor$. When searching from the root to a leaf on the last level, the search on Bob's tree is slower by

    A a factor of 2

    B a factor of 4

    C ** about 2 levels

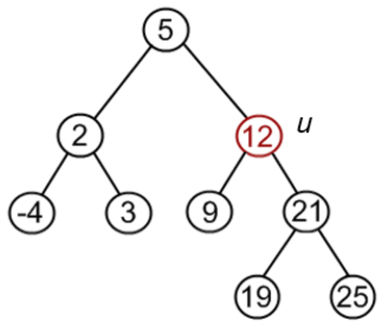
    D about 4 levels

    E there is no difference

11. The worst-case time of an insertion into a binary search tree containing n nodes and having height h is (tightest bound possible):

A ** $O(h)$ **B** $O(\log h)$ **C** $O(n)$ **D** $O(\log n)$

12. Key 12 in node u is deleted from the binary search tree shown. Which statement **cannot be true** after the deletion?



A The height of the tree does not change.

B ** The height of the tree reduces from 3 to 2.

C Node u contains key 9 and the node containing 9 is deleted.

D Node u contains key 19 and the node containing 19 is deleted.

13. A 2-3 tree has height k. After the insertion of a key, the tree has what height?

A k **B** $k-1$ **C** $k+1$ **D** ** A or C

14. T is a 2-3 tree of height 12. The search for key x can end **unsuccessfully** at a node on level 3.

A True **B** ** False

15. If one inserts n unique keys into an empty binary search tree in sorted order, the resulting tree will have height $\lceil \log n \rceil$.

A True **B** ** False

16. An insertion into a LL RB tree of size n makes a search towards a leaf and then makes how many rotations and color changes to balance the entire tree, in the worst case?

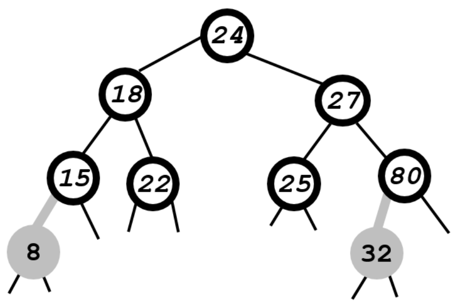
A O(1) left rotations, O(1) right rotations, and O(1) flip color operations

B $O(\log n)$ left rotations, $O(\log n)$ right rotations, and O(1) flip color operations

C ** $O(\log n)$ left rotations, $O(\log n)$ right rotations, and $O(\log n)$ flip color operations

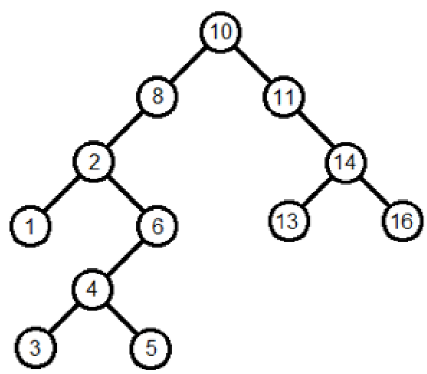
D $O(\log n)$ left rotations, $O(\log n)$ right rotations, and $O(n)$ flip color operations

17. Consider the LL BR tree T shown below (red nodes/links are shown in **grey**). How many insertions **not needing** any rotations or recoloring can be executed on T?



A none **B** 1 **C** ** 2 **D** 3 **E** 4

18. Consider the tree shown below. Visiting the nodes in a preorder traversal generates



A 16 14 13 11 10 8 6 5 4 3 2 1

B ** 10 8 2 1 6 4 3 5 11 14 13 16

C 1 2 3 4 5 6 8 10 11 13 14 16

D 10 8 11 2 14 1 6 13 16 4 3 5

19. A undirected graph on n vertices and m edges is represented by **adjacency lists.** We can determine whether the graph consists of exactly two connected components in time:

A $O(n)$

B $O(n^2)$

C ** $O(n+m)$

D $O(nm)$

20. A undirected graph G on n vertices and m edges is represented by an **adjacency matrix.** Running BFS on G starting at vertex 0, we can determine the number of edges on the shortest path from vertex 0 to vertex 2 in what time:

A $O(1)$

B $O(n)$

C $O(n+m)$

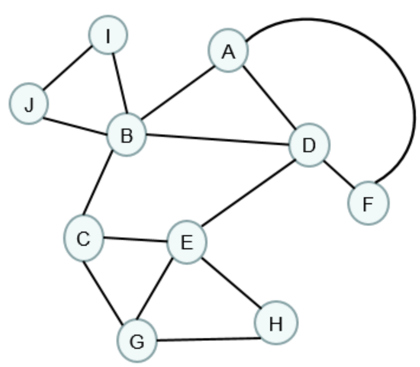
D ** $O(n^2)$

E $O(nm)$

21. Given a Merkle tree of height k, what is the **maximum number** of path siblings a leaf can have (excluding itself)?

A $\lceil \log k \rceil$ **B** $k-1$ **C** ** k **D** $k+1$

22. The graph shown below is represented by adjacency lists. In each list, vertices appear in increasing lexicographical order, When a DFS starts at vertex A, the vertices are marked in what order?



A A D F E H G C B J I

B A B C D E F G H I J

C A B I J C E G H D F

D ** A B C E D F G H I J