Quiz 2 Solutions

Problem 1. Trees

Q1 (10 points): Describe an $O(n)$ algorithm to make a **Balanced Binary Search Tree** from a sorted array of n elements

Solution:

From the sorted list, we want to create a balanced binary search tree. We can do this by setting the root as the middle element of the list, and letting the first half of the list be its left subtree and the second half be its right subtree (recursively creating the balanced subtrees as well). This is $O(n+n) = O(n)$ time.

Thus, the algorithm is $O(n)$

Q2 (5 points): What is the worst case running time to search for an element in an AVL tree with $n \cdot 2^n$ elements?

Solution:

The worst case search in a balanced binary search tree is $\Theta(h)$ where h is the height of the tree. Thus the running time for a search is

$$\begin{aligned} Time &= \Theta(\log(n \cdot 2^n)) \\ &= \Theta(\log(n) + \log(2^n)) \\ &= \Theta(\log(n) + n \cdot \log(2)) \\ &= \Theta(\log(n) + n) \\ &= \Theta(n) \end{aligned}$$

Problem 2. Hashing

Q1 (5 points): Draw the 11-item hash table (from slot 0 to slot 10) resulting from hashing the keys 12, 44, 13, 88, 23, 94, 11, 39, 20, 16, 5 using the hash function $h(i) = (3i + 4) \mod 11$ with linear probing. Note: i is the key.

Solution:

Slot	Key
0	94
1	39
2	16
3	5
4	44
5	88
6	11
7	12
8	23
9	20
10	13

Q2 (10 points): Given an array A with n elements and a number x, give an algorithm to find a pair of numbers in A whose sum is x. The algorithm should run in $O(n)$ time. Assume that a pair always exists in A. Justify the running time.

Solution:

```
1) Initialize an empty hash table s.
2) Do following for each element A[i] in A[]
   (a)    If s[x - A[i]] is set then print the pair (A[i], x - A[i])
   (b)    Insert A[i] into s
```

The algorithm has an amortised insertion cost of $O(1) \cdot O(n)$. It goes through the array A only once.

Problem 3. Short Answers

Q1 (7 points): Consider an undirected unweighted graph G. Let a breadth-first traversal of G be done starting from a node r. Let $d(r, u)$ and $d(r, v)$ be the lengths of the shortest paths from r to u and v respectively, in G. Is it true that $d(r, u) < d(r, v)$? Justify your answer.

Solution:

No. We cannot guarantee $d(r, u) < d(r, v)$. $d(r, u)$ and $d(r, v)$ will be equal when u and v are at same level, otherwise $d(r, u)$ will be less than $d(r, v)$. Hence, we can guarantee that $d(r, u) <= d(r, v)$

Q2 (8 points): How many undirected graphs (not necessarily connected) can be constructed out of a given set $V = \{V_1, V_2, \cdots, V_n\}$ of n vertices?

Solution:

In an undirected graph, there can be maximum $\frac{n\cdot(n-1)}{2}$ edges. We can choose to have (or not have) any of the $\frac{n\cdot(n-1)}{2}$ edges. So, total number of undirected graphs with n vertices is $2^{(n\cdot(n-1))/2}$.

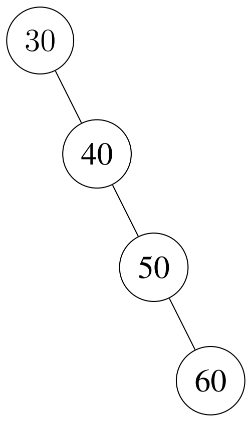
Problem 4. True/False

For each of the following questions, answer either True or False. **Explain your choice.**

Q1 (2 points): The worst case time complexity for search, insert and delete operations in a general Binary Search Tree is $O(\log n)$

Solution: **False**. In skewed Binary Search Tree (BST), all three operations can take $O(n)$.

See the following example:



Q2 (3 points): The minimum number of nodes in a binary tree of height h is $2^h - 1$. (The height of a tree with single node is considered to be 1)

Solution: **False**. A tree has maximum nodes if all levels have maximum nodes. So maximum number of nodes in a binary tree of height h is $1+2+4+\cdots+2^{h-1}$. The sum of this geometric series is $2^h - 1$. Thus, the maximum number of nodes in a binary tree of height h is $2^h - 1$.