Algorithms (50%)

1. (10%) Answer TRUE or FALSE for the following equation: $n^2 + n\lg n + \frac{1}{2}n = O(n^8)$

2. (15%) Solving the recurrence $T(n) = 2T(\lfloor\sqrt{n}\rfloor) + \lg n$ using big-O notation as tight as possible.

3. (10%) Consider the following two problems in which we are given a directed graph $G=(V, E)$ and vertices $u, v \in V$.
   Unweighted shortest path problem: Find a path from u to v consisting of the fewest edges.
   Unweighted longest simple path problem: Find a path from u to v consisting of the most edges.
   (a) (5%) Determine which problem can be solved using the dynamic-programming in polynomial time.
   (b) (5%) Determine which problem cannot be solved using the dynamic-programming in polynomial time, and also give the reason.

4. (15%) Given a sequence $K = \langle k_1, k_2, \ldots k_n \rangle$ of n distinct keys in sorted order such that $k_1 < k_2 < \cdots < k_n$, and we wish to build a binary search tree from these keys. For each key k_i, we have a probability p_i that a search will be for k_i. Some searches may be for values not in K, and so we also have $n+1$ "dummy keys" $d_0, d_1, d_2, \ldots, d_n$ representing values not in K. In particular, d_0 represents all values less than k_1, d_n represents all values greater than k_n, and for $i=1,2,\ldots,n-1$, the dummy key d_i represents all values between k_i and k_{i+1}. For each dummy key d_i we have a probability q_i that a search will correspond to d_i. Each key k_i is an internal node, and each dummy key d_i is a leaf. Every search is either successful (finding some key k_i) or unsuccessful (finding some dummy key d_i), and so we have $\sum_{i=1}^{n} p_i + \sum_{i=0}^{n} q_i = 1$. The expected cost of a search tree T is

$$E[\text{search cost in } T] = \sum_{i=1}^{n}(depth_T(k_i)+1)\cdot p_i + \sum_{i=0}^{n}(depth_T(d_i)+1)\cdot q_i =$$
$$1 + \sum_{i=1}^{n} depth_T(k_i)\cdot p_i + \sum_{i=0}^{n} depth_T(d_i)\cdot q_i,$$

where $depth_T$ denotes a node's depth in the tree T. Given five keys with $p_1 = 0.15, p_2 = p_4 = q_5 = q_1 = 0.10, p_3 = q_0 = q_2 = q_3 = q_4 = 0.05, p_5 = 0.20$, compute the corresponding smallest search cost.

本試題是否可以使用計算機： □可使用 ， ☑不可使用 （請命題老師勾選）

5. (10%) True/False Question

   a. A complete binary tree is also an AVL tree
   b. Radix sort can only be performed on sequential lists, not on linked lists
   c. For a complete binary tree represented in memory as an array, if there is a node at index 4i+3 it must be a child of a child (grandchild) of the node at i.
   d. When applied to an already sorted list, quick sort shows its worst-case complexity. When applied to a reverse-sorted list, quick sort shows its best-case complexity.
   e. Searching for a key in a heap takes worst-case time O(n).

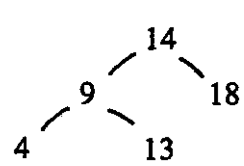
6. (15%) Given a binary search tree(BST), three traversals have been defined: *preorder, inorder,* and *postorder*. It returns the relative position of a node in the corresponding traversal. Given the following *preorder* traversal of a binary search tree

   8 2 1 4 6 5 16 32 24 27

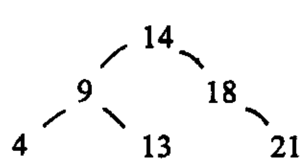
   List the results of the other two traversals and draw the corresponding BST tree as well.

7. (10%) Assume that the trees below are AVL trees. First inset a new node with a key of 12 into (a). Next, insert a new node with a key of 3 into (b), For both parts, show the trees before and after each rotation you perform.

(a)



(b)



8. (15%) Given a string *S*, and we determine if the string *S* satisfies the following conditions

   a. *S* contains repeated characters such as *xxxyy* form (*x* and *y* are characters)
   b. *S* contains *ABA'* form, where *A* is a sub-string containing characters different from sub-string *B*, and *A'* is the reverse form of *A*.

   Design the data structure and describe the procedure.