Q1 (20 points): Sort in Linear Time

(a) (6 points) Please describe any reason why we choose the counting sort algorithm to sort each digit in the Radix Sort?

(b) (6 points) Please describe the reason why the Counting Sort algorithm loops through the input array A backward to put elements into the output array B. That is, in line 9 of the pseudocode on page 108 of textbook, why it uses

```plaintext
for j <-- length[A] downto 1
```

rather than

```plaintext
for j <-- 1 to length[A]
```
(c)(8 points) If we would like to represent the decimal number $16^3 - 1$ as a radix-16 (hexadecimal) number, then how many digits this hexadecimal number has? what is the largest possible value for each digit?

• Q2(20 points): Stacks, Queues and Linked List

(a)(10 points) Describe how to use two queues to implement a stack so that push runs in $O(1)$ and pop runs in $O(n)$. Suppose the queues have no size limit. Please describe your algorithm without pseudocode.
(b)(10 points) Please write a pseudocode for \textbf{List-Insert(head, x, y)}. This procedure is to insert a new node $x$ before an existing node $y$ in a doubly non-circular list.

- **Q3(20 points): Hashing**
  Suppose we would like to insert a sequence of numbers into a hash table with table size 8 using the three open addressing methods, with the primary hash function $h_1(k) = k \mod 8$, the secondary hash function $h_2(k) = 1 + k \mod 7$, and the constants $c_1 = c_2 = 1/2$ (in quadratic probing).

  (a)(10 points) If the sequence of numbers is $<63, 37, 24, 29, 21>$, please successively insert these numbers into the following tables.

<table>
<thead>
<tr>
<th>index</th>
<th>linear</th>
<th>quadratic</th>
<th>double</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
(b) (3 points) For the hashing functions and table size we used in this question, does the linear probing fully utilize the table? How about the quadratic probing and double hashing?

(c) (7 points) A hash table with size 10 stores 6 elements. These 6 elements are stored in T[0], T[1], T[4], T[5], T[8], T[9]. Suppose that all the other entries contain no “deleted” flag. An entry has a “deleted” flag means that this entry stored an element before, but the element has already been deleted. If we would like to search an element with a key $k$ and assume the linear probing technique is used, what is the expected number of probes for an unsuccessful search?
• Q4(20 points): Binary Search Trees

For a given input array \( A : 9, 14, 17, 3, 8, 6, 10, 12, 5, 16 \),

(a)(5 points) What is the resulting binary search tree after inserting the numbers in the list to an initially empty tree?

(b)(5 points) From the tree you have built in part (a), what is the resulting tree after deleting the value 9?
(c) (10 points) Given a sorted array with \( n \) numbers, please describe an \( O(n) \) algorithm to construct a balanced binary search tree for these \( n \) numbers.
Q5 (20 points): B Trees

(a) (10 points) For a sequence of keys \{a, b, c, d, e, f, g, h, i, j\}, suppose we would like to construct a B-Tree, with degree 2, by successively inserting those keys one at a time, starting with an empty tree. Please draw the sequence of B-Trees after inserting each of the 10 keys.
(b)(5 points) Let $t$ be the degree of a BTree. Suppose the size of each object, including the key, stored in the tree is 128 bytes. Also, suppose the size of a BTreeNode pointer is 32 bytes. Suppose each BTreeNode has only a parent pointer, a list of objects, and a list of child pointers. What is the optimal degree for this BTree if a disk block is 4096 bytes?

(c)(5 points) Up to how many pieces of data a B-Tree, with degree 10 and with tree height 4 (5 levels), can hold?