List Problems

Zadid Hasan

February 2020

1 Union

1.1 Array Union

Suppose you are given two arrays A and B. You have to create another array C such that $C = A \cup B$. For example, $A = 1,2,2,5,8$ and $B = 1,3,4,6,8$. Here all of the numbers in the arrays will be less than 1000000. Here C will contain any value that occurs in either of the arrays. So, $C = 1,2,3,4,5,6,8$.

int [ ] Union(int A[ ], int B[ ])

1.2 Circular Array Union

In this case you will be given two circular arrays that are full. There is no empty space in either array. You have to find their union.

int [ ] Union(int A[ ], int startA, int B[ ], int startB)

1.3 Linked List Union

You will be given two singly non-dummy headed linear linked list. Find their union.

Node Union(Node A, Node B)

2 Intersection

2.1 Array Intersection

Suppose you are given two arrays A and B. You have to create another array C such that $C = A \cap B$. For example, A = 1,2,2,5,8 and B = 1,3,4,6,8. Here all of the numbers in the arrays will be less than 1000000. Here C will contain any value that occurs in both of the arrays. So, C = 1,8.

int [ ] Intersection(int A[ ], int B[ ])

2.2 Circular Array Intersection

Solve the same problem for full circular arrays.
int [ ] Intersection(int A[ ], int startA, int B[ ], int startB)

2.3 Linked List Intersection

Solve the above problem for two linked lists.
Node Intersection(Node A, Node B)

3 Sort

3.1 Array Sort

You are given an array A. For every $0 \leq i < A.length$, the following holds, $0 \leq A[i] < 1000000$. Sort the array as efficiently as possible.
int [ ] Sort(int A[ ])

3.2 Circular Array Sort

You are given a circular array A. For every $0 \leq i < A.length$, the following holds, $0 \leq A[i] < 1000000$. Sort the array as efficiently as possible.
int [ ] Sort(int A[ ],int start)

3.3 Linked List Sort

You are given a linked list A of integers. For every element of the linked list is between 0 and 999999. Sort the linked list as efficiently as possible.
Node Sort(Node A)

4 ZigZag

Suppose you are given two arrays A and B of the same size. You have to create another new array C such that the 1st element of C is the 1st element of A and the 2nd element of C is the 1st element of B and the 3rd element of C is the 2nd element of A and the 4th element of C is the 2nd element of B and so on. For example, if A = 1, 3, 5, 7 and B = 2, 4, 6, 8, then C will be 1, 2, 3, 4, 5, 6, 7, 8.
int [ ] ZigZag(int A[ ], int B[ ])
Also, solve the same problem for circular arrays and linked list.

5 Product

You will be given two linked lists of the same size. They will contain integers. You will find the product list of the two lists. For example if A = 1,2,3 and B = 4,5,6, then C will be 4, 10, 18.

Node Product(Node A, Node B)

6 Sum

You will be given two linked lists of the same size. They will contain integers. You will find the sum list of the two lists. For example if A = 1,2,3 and B = 4,5,6, then C will be 5, 7, 9.

Node Sum(Node A, Node B)

7 Difference

You will be given two linked lists of the same size. They will contain integers. You will find the difference list of the two lists. For example if A = 4,8,2 and B = 3,5,6, then C will be 1, 3, -4.

Node Difference(Node A, Node B)

8 Dot Product

You will be given two linked lists of the same size. They will contain integers. You will find the dot-product of the two lists. For example if A = 1,2,3 and B = 4,5,6, then dot product will be 4+10+18 = 32. Dot product for two lists is defined as follows,

$$DP(A, B) = \sum_{i=1}^{n} A_i B_i$$

int DotProduct(Node A, Node B)

9 Reverse Concatenation

Suppose you are given two lists A and B. First reverse B and then reverse A and append reversed A after reversed B.

Node RC(Node A, Node B)

10 Insertion in a Sorted List

Suppose you are given a sorted Array A. And the array has n elements. And you are given a number m to be inserted into A such that A remains sorted after the insertion.

int [ ] SortedInsertion(int A[ ], int n, int m)

11 Frequency of Numbers in a List

Suppose you are given an array A where all the elements are positive and smaller than 1000000. For every unique number, print it's frequency.

void Frequency(int A[ ])

12 Second Max of a List

Suppose you are given an array or a linked list of integers. Find the second highest value of that list.

int SecondMax(int A[ ])

13 Median Of a List

Suppose you are given an array of numbers. This array has odd numbers of integers. The median is the value in the middle of the array after the array is sorted. Find the median of that array.

int Median(int A[ ])

14 toList

Suppose you are given an array. Turn it into a linked list.

Node toList(int A[ ])

15 toArray

Suppose you are given a linked list. Turn this into an array.

int [ ] toArray(Node A)

16 Merge Two Sorted Arrays

Suppose you are given two sorted arrays A and B. Merge the two arrays in such a way that the resulting array remains sorted.

int [ ] mergeSorted(int A[ ],int B[ ])