```c
#include <stdio.h>

#include <stdlib.h>

#include <time.h>


struct Node{

    int x;

    struct Node *prev;

    struct Node *next;

};




struct Node* node_new(int x)

{

    struct Node *p;

    p = malloc(sizeof(struct Node));

    p->x = x;

    p->next = NULL;

    p->prev = NULL;


    return p;

}


struct Node* node_ins(struct Node *r, int x)

{

    if(r==NULL)

        return node_new(x);
    else if(x < r->x)
        r->prev = node_ins(r->prev, x);
    else
        r->next = node_ins(r->next,x);
    return r;
}

struct Node* node_min(struct Node *r)
{
    if(r == NULL)
        return NULL;
    else if(r->next != NULL)
        return node_min(r->next);
    return r;
}

struct Node* node_del(struct Node *r, int x)
{
    if(r==NULL)
        return NULL;
    if (x < r->x)
        r->prev = node_del(r->prev, x);
    else if(x > r->x)
        r->next = node_del(r->next, x);
    else{
        if(r->next==NULL && r->prev==NULL){
            free(r);
            return NULL;

        }

        else if(r->next==NULL || r->prev==NULL){

            struct Node *temp;

            if(r->next==NULL)

                temp = r->prev;

            else

                temp = r->next;

            free(r);

            return temp;

        }

        else

        {

            struct Node *temp = node_min(r->prev);

            r->x = temp->x;

            r->prev = node_del(r->prev, temp->x);

        }

    }

    return r;

}


struct Node* node_search(struct Node *r, int x)

{

    if(r==NULL || r->x==x)

        return r;

    else if(x < r->x)

        return node_search(r->prev, x);
    else
        return node_search(r->next,x);
}

void node_array_print(int arr[],int siz){
    printf("Normal Numbers : ");
    int i;
    for(i=0; i<siz; i++){
        printf("%d ",arr[i]);
    }
    printf("\n");
}

void node_tree_print( struct Node* r, int sp )
{

  int i;
  if( r != NULL )
  {
    node_tree_print( r->next, sp + 3 );
    for( i = 0; i < sp; i++ )
        printf(" ");
    printf("%d\n",r->x );
    node_tree_print( r->prev, sp + 3 );
  }

}
```

```
void node_sort(struct Node *r)

{

   if(r!=NULL){

      node_sort(r->prev);

      printf("%d ", r->x);

      node_sort(r->next);

   }

}


struct Node *node_add_auto(struct Node *r){

           int i, arr[10];

           for(i=0; i<10; i++){

                      arr[i] = rand()%100+1;

           }

           node_array_print(arr,10);

           r = node_new(arr[0]);


           for(i=1; i<10; i++){

                      node_ins(r,arr[i]);

           }

           printf("\n-Doubly Linked List Tree-------------\n\n");

           node_tree_print(r,0);

           printf("\n-----------------------------------\n");

           return r;

}
```

```
int main()

{

	srand(time(0));

	struct Node *r = NULL;


	r = node_add_auto(r);

	printf("\n");


	printf("Sort Numbers  : ");

	node_sort(r);


	printf("\n\n");


   return 0;

}
```