Nithin S
221IT085

IT206 Lab Assignment

Q1) Implement Heap Sort

CODE

```
#include<stdio.h>
#include<stdlib.h>

void swap(int *a,int*b)
{
int temp=*a;
*a=*b;
*b=temp;
}

void maxHeapify(int A[],int n,int i)
{
int largest=i;
int left=2*i+1;
int right=2*i+2;

if(left<n && A[largest]<A[left]) largest=left;
if(right<n&&A[largest]<A[right]) largest=right;
if(largest!=i)
{
swap(&A[i],&A[largest]);
maxHeapify(A,n,largest);
}

}
```

```
void buildMaxHeap(int A[],int n)
{
int lastNonLeaf=n/2-1;
for(int i=lastNonLeaf;i>=0;i--) maxHeapify(A,n,i);
}

void heapSort(int A[],int n)
{
buildMaxHeap(A,n);
for(int i=n-1;i>0;i--)
{
swap(&A[i],&A[0]);
n--;
maxHeapify(A,n,0);
}
}

int main()
{
int n;
printf("Enter the number of elements in the array:
");
scanf("%d",&n);
int *A=(int*)malloc(sizeof(int)*n);
printf("Enter the %d elements of the array: ",n);
for(int i=0;i<n;i++) scanf("%d",&A[i]);
heapSort(A,n);
printf("The sorted array:\n");
for(int i=0;i<n;i++) printf("%d ",A[i]);
return 0;
}
```

OUTPUT

```
student@HP-Elite600G9-08:~/Desktop/assgn$ gcc heapSort.c
student@HP-Elite600G9-08:~/Desktop/assgn$ ./a.out
Enter the number of elements in the array: 12
Enter the 12 elements of the array: 2 33 876 12 4 677 23 65 12 90 09 65
The sorted array:
2 4 9 12 12 23 33 65 65 90 677 876 student@HP-Elite600G9-08:~/Desktop/assgn$ 6~
```

Q2) Implement Priority Queue

CODE

```
#include<stdio.h>
#include<stdlib.h>

void swap(int *a,int*b)
{
int temp=*a;
*a=*b;
*b=temp;
}

void maxHeapify(int A[],int n,int i)
{
int largest=i;
int left=2*i+1;
int right=2*i+2;

if(left<n && A[largest]<A[left]) largest=left;
if(right<n&&A[largest]<A[right]) largest=right;
if(largest!=i)
{
swap(&A[i],&A[largest]);
maxHeapify(A,n,largest);
}

}

void buildMaxHeap(int A[],int n)
{
int lastNonLeaf=n/2-1;
for(int i=lastNonLeaf;i>=0;i--) maxHeapify(A,n,i);
}

void insert(int A[],int *n,int item)
{
(*n)++;
```

```
int i=*n-1;
A[i]=item;
while (i > 0 && A[(i - 1) / 2] < A[i])
{
swap(&A[i], &A[(i - 1) / 2]);
i = (i - 1) / 2;
}
}

int extractMax(int A[],int *n)
{

if(*n<=0) return -1;
if(*n==1)
{
int x=A[0];
*n--;
return x;
}
int max=A[0];
A[0]=A[*n-1];
(*n)--;
maxHeapify(A,*n,0);
return max;
}

void increaseKey(int A[], int index, int newValue) {
if (newValue < A[index]) {
printf("New value is smaller than the current value.
Cannot increase key.\n");
return;
}
A[index] = newValue;
maxHeapify(A, index + 1, index);
}



int main()
{
int n;
```

```
printf("Enter the number of elements in the array:
");
scanf("%d",&n);
int *A=(int*)malloc(sizeof(int)*n);
printf("Enter the %d elements of the array: ",n);
for(int i=0;i<n;i++) scanf("%d",&A[i]);
printf("Max Heap: ");
buildMaxHeap(A,n);
for(int i=0;i<n;i++) printf("%d ",A[i]);
printf("\n");

int choice, item;
while (1) {
printf("Choose operation:\n");
printf("1. Insert\n");
printf("2. Extract Max\n");
printf("3. Quit\n");
scanf("%d", &choice);

switch (choice) {
case 1:
printf("Enter element to insert: ");
scanf("%d", &item);
insert(A, &n, item);
printf("Inserted %d.\n", item);
printf("The Array now is: ");
for (int i = 0; i < n; i++) printf("%d ", A[i]);
break;

case 2:
item = extractMax(A, &n);
if (item == -1) {
printf("Priority queue is empty.\n");
} else {
printf("Extracted Max: %d\n", item);
}
printf("The Array now is: ");
for (int i = 0; i < n; i++) printf("%d ", A[i]);
break;

case 3:
```

```
free(A);
return 0;

default:
printf("Invalid choice.\n");
}
}
return 0;
}
```

OUTPUT

```
student@HP-Elite600G9-08:~/Desktop/assgn$ gcc priorityQueue.c
student@HP-Elite600G9-08:~/Desktop/assgn$ ./a.out
Enter the number of elements in the array: 5
Enter the 5 elements of the array: 1 2 3 4 5
Max Heap: 5 4 3 1 2
Choose operation:
1. Insert
2. Extract Max
3. Quit
1
Enter element to insert: 56
Inserted 56.
The Array now is: 56 4 5 1 2 3 Choose operation:
1. Insert
2. Extract Max
3. Quit
2
Extracted Max: 56
The Array now is: 5 4 3 1 2 Choose operation:
1. Insert
2. Extract Max
3. Quit
1
Enter element to insert: 6
Inserted 6.
The Array now is: 6 4 5 1 2 3 Choose operation:
1. Insert
2. Extract Max
3. Quit
2
Extracted Max: 6
The Array now is: 5 4 3 1 2 Choose operation:
1. Insert
2. Extract Max
3. Quit
3
student@HP-Elite600G9-08:~/Desktop/assgn$
```