**Assignment-2**

**CS1105 (Design and Analysis of Algorithm)**



**FACULTY GUIDE:**

Mr Devendra Bhavsar

Dr. Renu Jain

**SUBMITTED BY:** Siddhi Nyati

**Roll No:** 2022Btech101

**An interactive program for Priority Queue**

*CODE:*

#include <iostream>

#include <vector>

using namespace std;

struct Process {

int id;

int priority;

};

class PriorityQueue {

private:

vector<Process> heap;

void heapifyUp(int index) {

while (index > 0) {

int parentIndex = (index - 1) / 2;

if (heap[index].priority > heap[parentIndex].priority) {

swap(heap[index], heap[parentIndex]);

index = parentIndex;

} else {

break;

}

}

}

void heapifyDown(int index) {

int leftChild = 2 \* index + 1;

int rightChild = 2 \* index + 2;

int largest = index;

if (leftChild < heap.size() && heap[leftChild].priority > heap[largest].priority) {

largest = leftChild;

}

if (rightChild < heap.size() && heap[rightChild].priority > heap[largest].priority) {

largest = rightChild;

}

if (largest != index) {

swap(heap[index], heap[largest]);

heapifyDown(largest);

}

}

public:

void insertProcess() {

Process newProcess;

cout << "Enter process ID: ";

cin >> newProcess.id;

cout << "Enter priority: ";

cin >> newProcess.priority;

heap.push\_back(newProcess);

heapifyUp(heap.size() - 1);

cout << "Process with ID " << newProcess.id << " inserted with priority " << newProcess.priority << endl;

}

void deleteProcess() {

if (heap.empty()) {

cout << "Heap is empty. Nothing to delete." << endl;

return;

}

Process deletedProcess = heap[0];

heap[0] = heap.back();

heap.pop\_back();

heapifyDown(0);

cout << "Process with ID " << deletedProcess.id << " and priority " << deletedProcess.priority << " deleted." << endl;

}

void displayProcesses() {

if (heap.empty()) {

cout << "Heap is empty." << endl;

} else {

cout << "Processes in the heap:" << endl;

for (const Process &p : heap) {

cout << "ID: " << p.id << ", Priority: " << p.priority << endl;

}

}

}

};

int main() {

PriorityQueue priorityQueue;

while (true) {

cout << "\nChoose an operation:" << endl;

cout << "1. Insert\n2. Delete\n3. Display\n4. Exit\n";

int choice;

cin >> choice;

switch (choice) {

case 1:

priorityQueue.insertProcess();

break;

case 2:

priorityQueue.deleteProcess();

break;

case 3:

priorityQueue.displayProcesses();

break;

case 4:

cout << "Exiting the program." << endl;

return 0;

default:

cout << "Invalid choice. Please choose again." << endl;

}

}

return 0;

}

*OUTPUT:*  
Choose an operation:

1. Insert

2. Delete

3. Display

4. Exit

1

Enter process ID: 101

Enter priority: 3

Process with ID 101 inserted with priority 3

Choose an operation:

1. Insert

2. Delete

3. Display

4. Exit

1

Enter process ID: 201

Enter priority: 2

Process with ID 201 inserted with priority 2

Choose an operation:

1. Insert

2. Delete

3. Display

4. Exit

3

Processes in the heap:

ID: 101, Priority: 3

ID: 201, Priority: 2

Choose an operation:

1. Insert

2. Delete

3. Display

4. Exit

2

Process with ID 101 and priority 3 deleted.

Choose an operation:

1. Insert

2. Delete

3. Display

4. Exit

3

Processes in the heap:

ID: 201, Priority: 2

Choose an operation:

1. Insert

2. Delete

3. Display

4. Exit

2

Process with ID 201 and priority 2 deleted.

Choose an operation:

1. Insert

2. Delete

3. Display

4. Exit

3

Heap is empty.

Choose an operation:

1. Insert

2. Delete

3. Display

4. Exit

4

Exiting the program.