

**NAME : WAJAHAT SAFEER**

**CLASS : BSSE 3RD**

**SECTION : C**

**ROLL NO : 12407**

**SUBJECT : DATA STRUCUTURE AND ALGORITHIM**

**TEACHER : SIR JAMAL ABDUL AHAD**

**ASSIGNMENT : 2**

**Question 1 :**

**Design a python program that stimulates a web server handling incomming request using a queue. Model different types of request with various processing times and stimulate their processing order.**

**Answer :**

import queue

import time

class WebServer:

def \_\_init\_\_(self):

self.request\_queue = queue.Queue()

def add\_request(self, request):

print(f"Incoming request {request['type']} with processing time {request['processing\_time']} seconds.")

self.request\_queue.put(request)

def process\_requests(self):

while not self.request\_queue.empty():

request = self.request\_queue.get()

print(f"Processing {request['type']} request...")

time.sleep(request['processing\_time'])

print(f"{request['type']} request processed.")

web\_server = WebServer()

# Simulate incoming requests

requests = [

{'type': 'A', 'processing\_time': 3},

{'type': 'B', 'processing\_time': 5},

{'type': 'C', 'processing\_time': 2},

]

for request in requests:

web\_server.add\_request(request)

web\_server.process\_requests()

**Question 3 :**

**Discuss the time complexity off enqueue and dequeue opreations in basic queue. How can you optimize these operations for special use cases?**

**Answer:**

The time complexity of enqueue and dequeue operations in a basic queue depends on the implementation of the queue. If the queue is implemented using an array, the time complexity of both operations is O(1) ¹³. If the queue is implemented using a linked list, the time complexity of both operations is O(1) as well ².

To optimize these operations for specific use cases, you can use a circular queue. A circular queue is a variation of a queue data structure in which the last element is connected to the first element to make a circle. This makes it possible to use the spaces that are left empty after a dequeue operation. In a circular queue, the enqueue and dequeue operations can be performed in O(1) time complexity ¹.

Another way to optimize these operations is to use a priority queue. A priority queue is a type of queue in which each element is assigned a priority and is dequeued based on its priority. The elements with higher priority are dequeued first. The time complexity of enqueue and dequeue operations in a priority queue depends on the implementation of the priority queue. If the priority queue is implemented using a heap, the time complexity of both operations is O(log n) ³.

In conclusion, the time complexity of enqueue and dequeue operations in a basic queue is O(1) for both array and linked list implementations. To optimize these operations for specific use cases, you can use a circular queue or a priority queue.

Source: Conversation with Bing, 11/12/2023

(1) Time and Space Complexity Analysis of Queue operations. https://www.geeksforgeeks.org/time-and-space-complexity-analysis-of-queue-operations/.

(2) Queue Data Structure and Implementation in Java, Python and C/C++. https://www.programiz.com/dsa/queue.

(3) Queue - Linked List Implementation - GeeksforGeeks. https://www.geeksforgeeks.org/queue-linked-list-implementation/.

(4) What is the time complexity of enqueue and dequeue of a queue .... https://cs.stackexchange.com/questions/105029/what-is-the-time-complexity-of-enqueue-and-dequeue-of-a-queue-implemented-with-a.

**Question 4 :**

**How can you use two stacks to implement a queue? Provide a step by step explaination of the enqueue and dequeue opreations in this scenario.**

**Answer:**

You can use two stacks to implement a queue. Let’s call the two stacks `stack1` and `stack2`. Here are the steps for enqueue and dequeue operations:

\*\*Enqueue Operation:\*\*

1. Push the new element onto `stack1`.

\*\*Dequeue Operation:\*\*

1. If both `stack1` and `stack2` are empty, return an error message.

2. If `stack2` is empty, pop all elements from `stack1` and push them onto `stack2`.

3. Pop the top element from `stack2` and return it.

Here’s how the enqueue and dequeue operations work in this scenario:

1. Suppose we want to enqueue the element `5` into the queue. We push `5` onto `stack1`.

2. Suppose we want to enqueue the element `7` into the queue. We push `7` onto `stack1`.

3. Suppose we want to dequeue an element from the queue. We check if `stack2` is empty. Since it is empty, we pop all elements from `stack1` and push them onto `stack2`. Now `stack2` contains the elements in reverse order, so the top element of `stack2` is the first element that was pushed onto `stack1`, which is `5`. We pop `5` from `stack2` and return it.

4. Suppose we want to enqueue the element `9` into the queue. We push `9` onto `stack1`.

5. Suppose we want to dequeue an element from the queue. We check if `stack2` is empty. Since it is not empty, we pop the top element from `stack2` and return it. The top element of `stack2` is `7`, which was the second element that was pushed onto `stack1`.

Using two stacks to implement a queue can be useful in certain situations, such as when you have limited memory available or when you need to implement a queue using a stack data structure.

Source: Conversation with Bing, 11/12/2023

(1) Queue using Stacks – GeeksforGeeks. <https://www.geeksforgeeks.org/queue-using-stacks/>.

(2) How to Implement a Queue Using Two Stacks – Better Programming. <https://betterprogramming.pub/how-to-implement-a-queue-using-two-stacks-80772242b88c>.

(3) Implementation of Queue using two stacks in C – PrepInsta. <https://prepinsta.com/c-program/implementation-of-queue-using-two-stacks/>.

(4) How to Implement a Queue With Two Stacks – Stack Overflow. <https://stackoverflow.com/questions/34099279/how-to-implement-a-queue-with-two-stacks>.

(5) How to implement a queue using two stacks? – Stack Overflow. https://stackoverflow.com/questions/69192/how-to-implement-a-queue-using-two-stacks.