1. 자료구조 팀프로젝트 음식점 시나리오

2. 목차 시나리오소개 , 원형큐와 연결리스트큐 설명 , 원형큐로 만든 식당 , 연결리스트큐로 만든 식당 5. 마무리

3. 시나리오 소개

4. 음식점 자리시스템을 구현해 이를 비교해보려합니다.

기존에 배웠던 원형큐 은행업무를 활용하였으며 이를 연결리스트큐로 구현하여 차이가 발생하는 것을 확인하려합니다.

주된 기능은 음식점자리와 대기시간입니다.

부가적 기능은 주차장 과 대기줄입니다.

여기서 주차장은 정렬을 활용하였습니다. 대기줄은 원형큐로 구성되었습니다.

5.원형큐와 연결리스트큐에 대한 간단한 소개입니다.

6. //원형큐와 연결리스트큐에 대한 소개

잘알다시피 원형 큐는 선형 큐의 문제점을 보완하기 위한 자료구조입니다. 선형큐의 문제점은 rear이 가르키는 포인터가 배열의 마지막 인덱스를 가르키고 있을 때 앞쪽에서 Dequeue로 발생한 배열의 빈 공간을 활용 할 수가 없었습니다. 원형큐에서는 포인터 증가 방식이 (rear+1)%arraysize 형식으로 변환하기 때문에 배열의 첫 인덱스부터 다시 데이터의 삽입이 가능해집니다.

연결리스트큐에 경우 큐는 자료를 한쪽으로 보관하고 다른쪽에서 꺼내는 FIFO(First In First Out) 방식의 자료구조입니다. 큐에 자료를 보관하는 연산을 PUT 혹은 ENQUEUE라 말하고 꺼내는 연산을 GET 혹은 DEQUEUE라고 말합니다. 그리고 보관할 위치 정보를 rear, 꺼낼 위치 정보를 front라고 말합니다.

7. 원형큐로 구현된 식당시스템과 더하여 주차장 시스템에 대해 소개하겠습니다.

8. 원형큐로 대기시간과 자리를 구현하였습니다. 간단한코드는 (visual studio)

if (two\_time > 0) //2번테이블사용중

{

printf("손님 %d 2번테이블 사용중\n", two\_customer);

two\_time--;

if (two\_time == 0)

{

printf("(2번테이블 %d분 부터 사용가능합니다.)\n", clock + 1);

twoCounter = true;

}

}

else if (twoCounter)

{

if (!is\_empty(&q))

{

element customer = dequeue(&q);

two\_customer = customer.id;

two\_time = customer.service\_time;

printf("손님 %d이 %d분에 2번 테이블에서 식사를 시작합니다. 예상식사시간은 %d분입니다.\n", customer.id, clock, clock - customer.in\_time);

twoCounter = false;

total\_wait += clock - customer.in\_time;

}

}

테이블 구성 예시

이렇게 보이며 수업시간에 배운 은행업무를 활용해 총 자리는 3개로 만들어 졌습니다. 이렇게 코드를 구현하니 자리가 추가될때마다 코드가 반복되고 이에 따라 속도가 느려지는 것을 확인할 수 있었습니다.

추가적으로 전체 대기시간과 방문한 총 고객수를 확인할 수 있습니다.

9. 다음으로는 입출차 목록입니다.

입출차된 차들을 확인할 수 있으며 랜덤으로 저장된 큐에서 차량번호와 고객번호를 주고 이를 저장해 활용합니다.

10. 차량번호의 내림차순과 오름차순에 대한 코드입니다.

11. 고객번호의 대한 내림차순과 오름차순 입니다.

12. 연결리스트 큐에 대해 설명하겠습니다.

13. 우선 연결리스트큐로 구현한 식당파트 이전에 식당대기열을 원형큐로 따로 구현하여 자리와 대기시간이외 대기순서파트를 추가하였습니다. 대기순서는 자리와 대기시간과는 별개로 자리에 들어오기전 매장 밖에서 기다리는 손님들을 카운팅하기 위해 만들어졌습니다.

Visual code

printf("현재 식당 대기 순서 = ", q->front, q->rear);

if (!is\_empty(q)) {

int i = q->front;

do {

i = (i + 1) % (MAX\_QUEUE\_SIZE);

printf("%d | ", q->data[i]);

if (i == q->rear)

break;

} while (i != q->front);

}

printf("\n\n");

해당하시는 파트를 보시면 배웠던 원형큐를 활용해 대기순번에 랜덤한 고객을 추가시키고 이를 확인 시키는 것을 알 수 있습니다.

큐의 데이터가 delete되는 것을 손님이 매장내로 들어간 것을 표현한것입니다.

14.

15.

16. 감사합니다.