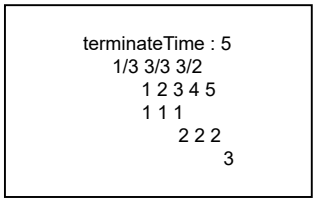
A




W

ServiceNo

Q

| int arrivalTime; | int arrivalTime; | int arrivalTime; | int arrivalTime; | int arrivalTime; |
|---|---|---|---|---|
| int serviceTime; | int serviceTime; | int serviceTime; | int serviceTime; | int serviceTime; |
| int startTime = 0; | int startTime = 0; | int startTime = 0; | int startTime = 0; | int startTime = 0; |
| int endTime = 0; | int endTime = 0; | int endTime = 0; | int endTime = 0; | int endTime = 0; |

Front

processArrival *currWaitTime: 현재 고객이 기다려야 하는 시간

```
if (currentTime == newCustomerNode->customer_data.arrivalTime)
```

```
if ((*currWaitTime) + currentTime >= terminateTime)
```

현재 시간과 도착 시간이 같고 기다려야 하는 시간 + 현재시간 < 종료 시간 즉 (작을때만) AQ -> WQ 로 노드를 보냅니다.

Rear

**curr**

currWaitTime + currentTime < termin
AQ -> WQ 로 노드 보내면서 *currwaitTime

```
*currWaitTime += newCustomer
```

시간이 지남에 따라 대기 시간 1씩 감소 하
대기 시간이 0 일땐 -- 를 하지 않는다.

```
if (currWaitTime > 0) currWa
```

WQ

| int arrivalTime; |
|---|
| int serviceTime; |
| int startTime = 0; |
| int endTime = 0; |

Front

processServiceNodeStart

```
int service = nextCustomerNode->customer_data.serviceTime;
nextCustomerNode->customer_data.startTime = currentTime;
nextCustomerNode->customer_data.endTime = currentTime + service;
nextCustomerNode->customer_data.status = start;
```

start = currenttime
end = current + service
status = start

ode

processServiceNodeEnd

1.ProcessServniceNodeEnd 의 반환값이 NULL일때 → 1-1.ServiceNode 가 비어있을때
1-2. Servic가 끝난 경우 (free하고 NULL

```
if (pServiceNode->cust
```

서비스가 끝날때마다 고객이 대기 했던 시
고객수도 끝날때마다 더

```
*pTotalW
(*pServi
```

2.ProcessServniceNodeEnd 의 반환값이 NULL이 아닐때 즉
끝나는 시간과 현재 시간이 다를때 노드를 그대로 반환 (~=서비스 중이다)

```
if (pServiceNode->customer_data.endTime != currentTime)
                    return (pServiceNode);
```

**ServiceNode != NULL 인 상태로 업무 시간이 마감되는경우**

**고객님이 서비스를 받다가 은행 시간이 종료되는 경우도 WaitTime 이 있으니 TotalWaitTime 에 더해줍니다.**

```
if (time >= terminateTime)
{
  if (serviceNode != NULL)
  {
    int arrival = serviceNode->customer_data.arrivalTime;
    int start = serviceNode->customer_data.startTime;
```

**rWaitTime 계산**

nateTime 작을때
+= service 더해준다.

```
rNode->customer_data.serviceTime;
```

는것.

```
aitTime--;
```

반환)

```
omer_data.endTime == currentTime)
```

간을 계속해서 더해준다.
해준다.

```
WaitTime += (start - arrival);
ceUserCount)++;
```

```
    TotalWaitTime += (start - arrival);
  }
  free(serviceNode);
  break;
}
```