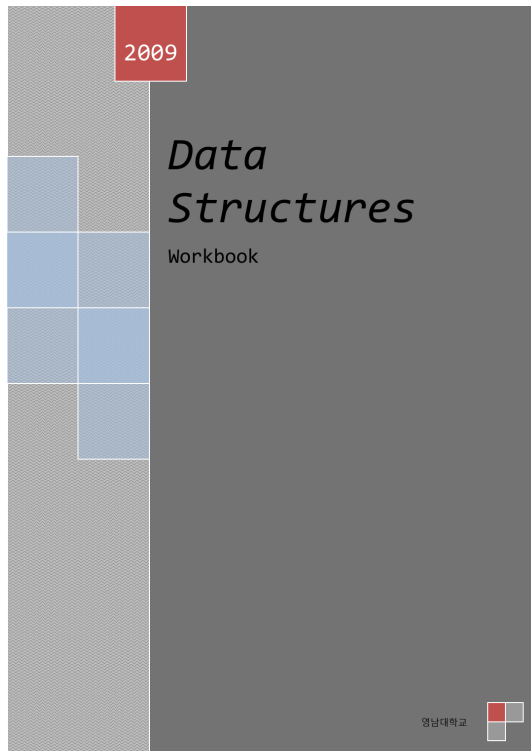


## • 영남대 NEXT사업 “Data Structure Workbook” 저작

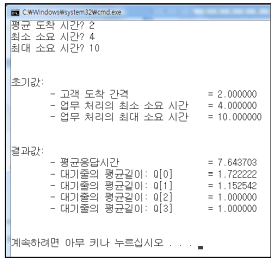
- 개요: 학부생이 어려워하는 ‘자료구조’ 과목을 혼자서도 공부할 수 있게 제작한 참고서.



영남대학교

Data Structures Workbook	3
목차	
Data Structures Workbook	2
목차	3
표, 그림	12
그림 목차	12
표 목차	14
1 장. 기본 개념	15
1. 개념정의	15
Computer Program ?	15
System Life Cycle의 개념	15
알고리즘 명세	16
재귀 알고리즘(Recursive Algorithms)	16
데이터 추상화(Data Abstraction)	17
추상적 데이터 타입(Abstract Data Type)	17
연산 명세에서 내부 함수들의 종류	17
ADT의 예: Natural Number	18
성능 분석(Performance Analysis)	18
공간 복잡성(Space Complexity)	19
시간 복잡성(Time Complexity)	19
근사 표현( $O$ , $\Omega$ , $\Theta$ )	19
근사 표현(1): $f(n) = O(g(n))$	20
근사 표현(2): $f(n) = \Omega(g(n))$	20
근사 표현(3): $f(n) = \Theta(g(n))$	20
성능 측정(Performance Measurement)	21
2. 기출문제	22
3. Sample Program: 행렬의 합과 곱	25
1) 문제 설명	25
2) 자료구조 설명	27

Data StructureWorkbook

3. SAMPLE PROGRAM: 은행업무 시뮬레이션	
1) 문제 설명	
4 명의 직원이 근무하는 은행을 가정하자. 시간 $t_0$ 에 은행에 도착한 고객 C 는 직원에게 은행 업무를 요청하게 되는데, 은행 업무 처리에 소요되는 기간은 $t_i$ 라고 가정하자. 만약, 4 명의 직원 중에서 현재 작업이 없는 직원이 있다면 C 는 그 직원에게 즉시 업무를 요청하여 $t_i$ 의 시간 후에 은행을 떠날 수 있을 것이다. 그러나 모든 직원이 다른 고객의 업무를 처리 중이라면, 각 직원의 대기줄 중에서 가장 짧은 대기줄에 C 는 대기하게 되고, 앞서 대기중인 모든 고객의 처리가 완료된 후 C 의 작업을 처리할 수 있다. 이 경우, C 는 (대기줄에서의 소요 시간 + $t_i$ )의 시간만큼 은행에서 기다려야 한다. 그렇다면, 임의의 고객에 대해 은행에 도착한 후 은행을 떠날 때까지의 평균대기시간은 얼마인가? 단, 고객은 한 명씩 도착 간격(inter-arrival time)만큼의 시차를 두고 은행에 차례대로 도착한다고 가정한다.	
<ul style="list-style-type: none"><li>입력<ul style="list-style-type: none"><li>은행 도착 시간의 편차(inter-arrival time)</li><li>고객이 요청하는 은행 업무 처리의 최소 소요 시간(minimal duration)</li><li>고객이 요청하는 은행 업무 처리의 최대 소요 시간(maximal duration)</li></ul></li><li>출력<ul style="list-style-type: none"><li>고객당 은행에서의 평균대기시간</li><li>각 직원의 대기줄에서 대기하고 있는 고객 수의 평균</li></ul></li></ul>	
결과화면	

Data StructureWorkbook

136 Sample Program: 은행업무 시뮬레이션

2) 자료구조 설명

프로그램의 요구사항은 고객당 은행에서의 **평균대기시간**, 즉 **평균응답시간**을 구하는 것과 각 창구에서 **대기하고 있는 고객수의 평균**을 구하는 것이다. 여기서 평균대기시간을 구하기 위해 각 고객마다 대기시간을 구해야 한다. 대기시간은 은행에 들어온 시간부터 업무가 종료될 까지의 시간 간격이다. 평균대기시간은 모든 고객의 대기시간의 총합을 누적인 고객으로 나눈 값이다. 또한 해당 각 창구의 길이는 도착이벤트가 발생하고 은행창구에 줄을 서는 순간 해당창구의 길이이다. **대기하고 있는 고객수의 평균**은 “해당 창구를 이용하는 각 고객에 대한 창구 길이의 누적”을 “해당 창구를 이용한 모든 고객의 수”로 나눈 값이다.

$k$	마지막 고객의 번호 (전체고객의 수가 된다)
$a_n$	번호가 $n$ 인 고객이 도착하는 이벤트
$d_n$	번호가 $n$ 인 고객이 떠나가는 이벤트
$T(a_n)$	$n$ 번째 고객이 도착했을 때의 시간
$T(d_n)$	$n$ 번째 고객이 떠나갈 때의 시간
$Len(Q[m])$	$m$ 번째 창구 길이

이라고 정의할 때, 프로그램을 통해 구하고자하는 값은 각각 다음과 같다.

- 은행에서의 평균대기시간:

$$\left(\frac{1}{k}\right) \cdot \sum_{n=1}^k (T(d_n) - T(a_n))$$

- 각 직원의 대기줄에서 대기하고 있는 고객 수의 평균:

모든  $a_n$ 에 대하여,  $Q[m] \in a_n$ 이라면

$$\sum_{n=1}^k (Q[m] \in a_n \text{되는 순간의 } Len(Q[m])) \div (Q[m] \text{을 이용한 총 고객수})$$

(여기서  $m$ 은 상수)

앞서 설명한 것을 토대로 구하고자 하는 것에 대한 정보 저장 형태는 다음과 같다.

은행에서의 평균대기시간	<ul style="list-style-type: none"> <li>총 고객수</li> <li>도착이벤트가 일어난 시간</li> <li>종료이벤트가 발생한 시간</li> </ul>
이벤트 정보	<ul style="list-style-type: none"> <li>이벤트 종류를 저장하는 변수</li> <li>이벤트 발생 시간</li> <li>이벤트 소요 시간</li> </ul>
창구에 대한 정보 (각 직원의 대기줄에서 대기하고 있는 고객수의 평균)	<ul style="list-style-type: none"> <li>창구에 줄 서 있는 고객들의 정보 = 도착이벤트의 정보 (도착시간, 업무소요시간)</li> <li>각 창구를 이용한 총 고객수</li> <li>창구에 들어왔을 때의 창구 길이에 대한 누적</li> </ul>

Data Structure

Workbook

List (리스트)

137

위의 “이벤트 정보”를 저장하려 한다. 이벤트 시간에 따라 정보의 순서가 항상 변화하므로 자료구조 중 “리스트”를 써야한다. 그런데 배열로 구현한 리스트를 사용하면 정보 삽입에 따른 위치조정이 자주 발생될 것이다. 여기서 이벤트의 최대 개수도 미리 알 수 없다. 따라서 삽입 및 삭제에 따른 순서 조정이 자유로우며, 개수에 제한을 받지 않는(동적할당이 가능한) “링크드 리스트”를 써야한다.

일반적으로 창구는 줄을 서서 먼저 들어 오는 사람을 먼저 처리하고, 나중에 들어오는 사람을 후에 처리하는 것이 일반적이다. 따라서 큐(queue)를 쓰며, 위의 이벤트리스트를 구현하는 것과 마찬가지로 큐에 대한 최대길이를 알 수 없다. 따라서 “링크드 리스트로 구현된 큐”가 “창구에 대한 정보”에 적합한 자료구조 형태가 된다.

여기서 “각 직원의 대기줄에서 대기하고 있는 고객수의 평균”을 구하기 위해서 해당 창구를 이용한 고객수와 창구에 들어왔을 때의 창구 길이에 대한 누적을 저장해야 한다. 이것은 “창구에 대한 정보”와 같이 조합한다.

“은행에서의 평균대기시간”을 구하기 위해서는 “총 고객수”를 저장하기 위한 변수와 “고객이 도착할 때의 시간”, “고객이 떠날 때의 시간”이 필요하다. “고객이 도착할 때의 시간”은 창구에 대한 정보 중 “도착시간”을 이용하면 되고, “고객이 떠날 때의 시간”은 이벤트리스트에 등록된 “은행에서 떠나는 이벤트”에서 “이벤트 시간”을 이용하면 된다.

예제를 통하여 좀 더 필요한 자료구조 혹은 변수, 그리고 알고리즘에 대해서 알아보자.

현재 시간은 0 이며, 고객의 은행도착시간 간격이 2 이고, 은행 업무종료시간은 5 이다. 창구직원 1 한명일 때, 각 업무소요시간이 3, 2, 3 인 고객이 연속해서 3 명이 들어오면 다음과 같이 고객의 정보를 정할 수 있다.

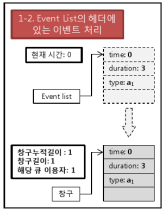
고객 1	고객 2	고객 3
이벤트	이벤트	이벤트
도착시간	도착시간	도착시간
업무소요시간	업무소요시간	업무소요시간

Data Structure

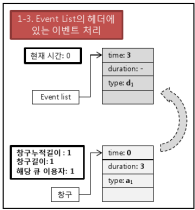
Workbook



과정 1-1 은행 시작과 동시에 고객 1에 대한 도착 이벤트( $a_1$ )가 발생한다. 이를 이벤트리스트에 등록하고 총고객수는 1이 된다. (실제 프로그램으로 구현 할때는 이벤트 종류에는 “몇 번째 고객이 도착하고 떠나간다”라는 것은 기록되지 않는다, 설명을 위해서 고객 번호를 같이 넣어준 것이다. 실제로는 이벤트 타입에는 “도착”과 “몇 번째 창구의 고객이 떠나 간다는 것”만 기록된다.)



과정 1-2 이벤트리스트에 등록된 이벤트에 대해 처리를 한다. 그러면 고객 1은 창구에 등록이 되고, 이때 해당 창구의 길이는 1이므로 누적을 해준다



과정 1-3 해당 창구의 첫번째 사용자는 3초 후에 업무가 종료되고, 그에 대한 종료 이벤트( $d_1$ )를 이벤트리스트에 등록한다.