|  |
| --- |
| **[511643] 자료구조 (2019-2학기)** |
| **실습 #11 보고서** |

|  |  |
| --- | --- |
| **이름** | 지현한 |
| **학번** | 20165164 |
| **소속**  **학과/대학** | 소프트웨어융합대학 빅데이터전공 |
| **분반** | 03 (담당교수: 김태운) |

## <주의사항>

* 개별 과제 입니다. (팀으로 진행하는 과제가 아니며, 모든 학생이 보고서를 제출해야 함)
* 각각의 문제 바로 아래에 답을 작성 후 제출해 주세요.
  + 소스코드/스크립트 등을 작성 한 경우, 해당 파일의 이름도 적어주세요.
* 스마트캠퍼스 제출 데드라인: 2019. 11. 20. (수요일) 23:59 // 2주 과제
  + 데드라인을 지나서 제출하면 24시간 단위로 20%씩 감점(5일 경과 시 0점)
  + 주말/휴일/학교행사 등 모든 날짜 카운트 함
  + 부정행위 적발 시, 원본(보여준 사람)과 복사본(베낀 사람) 모두 0점 처리함
  + 예외 없음
* 스마트캠퍼스에 아래의 파일을 제출 해 주세요
  + 보고서(PDF 파일로 변환 후 제출)
  + 보고서 파일명에 이름과 학번을 입력 해 주세요.
  + 소스코드, 스크립트, Makefile 등을 작성해야 하는 경우, 모든 파일 제출 (미 제출시 감점)

## <개요>

이번 과제는 해시테이블을 구현하는 과제입니다.

## <실습 과제>

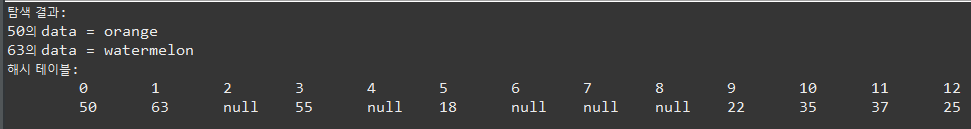
|  |
| --- |
| **[Q 0] 요약 [배점: 10]**  이번 과제에서 배운 내용 또는 과제 완성을 위해서 무엇을 했는지 2~3문장으로 요약하세요. |

답변: 해시 테이블에 대해 더 깊게 이해하게 되었고, 특징들을 잘 알게 되었습니다.

과제 완성을 위해서 강의 자료를 다시 찾아보았고, 강의자료에 나와있는 코드들을 이해 하려고 노력했습니다.

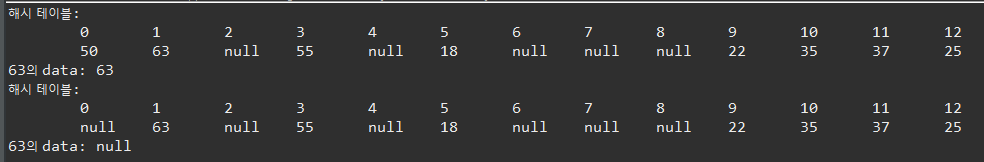
|  |
| --- |
| **[Q 1]** 선형조사방식, LinearProbing 클래스 [10점]  강의자료를 참고하여 LinearProbing 클래스를 구현하세요. 강의자료 p. 26과 동일하게 main 메소드를 구현하고 실행하세요. 강의자료에 나온 터미널 출력(= ‘해시 테이블’ 로 시작하는 출력 부분)과 동일하게 출력하는 print 메소드를 구현하세요. 터미널 출력 화면을 캡처해서 본 문서에 첨부하고, 소스코드도 제출하세요. |

답변: LinearProbing.java Q1main.java



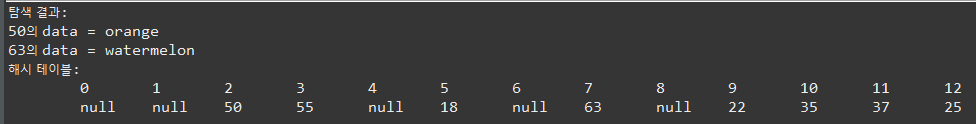
|  |
| --- |
| **[Q 2]** 선형조사방식 + delete 메소드, LinearProbing2 클래스 [20점]  LinearProbing 클래스의 코드를 최대한 재활용해서 LinearProbing2 클래스를 만드세요. LinearProbing2 클래스는 LinearProbing 클래스가 제공하는 모든 기능을 동일하게 제공해야 하며, public V delete(K key) 메소드를 포함하고 있어야 합니다 (삭제하는 항목의 데이터를 리턴함).   1. 강의자료 p.27과 같은 순서로 키 값을 추가하세요. K와 V 모두 Integer 형으로 설정하고, put 메소드를 호출할 때, data는 키 값과 동일하게 설정하세요. 2. 그리고 print 메소드를 호출하세요. 3. 키값이 63인 항목을 찾아서 터미널에 출력하세요 (터미널에는 get 메소드의 리턴값을 출력함) 4. 키 값이 50인 항목(a[0]에 저장된 항목)을 삭제하세요 5. print 메소드를 호출하세요 6. 키값이 63인 항목을 찾아서 터미널에 출력하세요. (터미널에는 get 메소드의 리턴값을 출력함)   터미널 출력 화면을 캡처해서 본 문서에 첨부하고, 소스코드도 제출하세요. |

답변: LinearProbing2.java Q2main.java



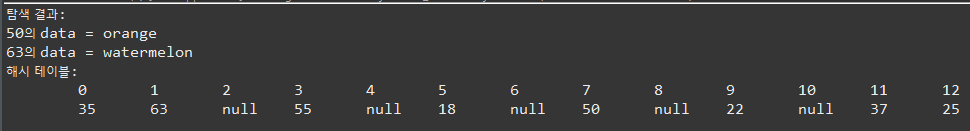
|  |
| --- |
| [Q3] 이차 조사 방식 [10점]  강의자료를 참고하여 이차조사방식 QuadProbing 클래스를 구현하세요. 강의자료 p.32와 같이 테스트를 위한 main 메소드를 구현하고 실행하세요.  터미널 출력 화면을 캡처해서 본 문서에 첨부하고, 소스코드도 제출하세요. |

답변: QuadProbing.java Q3main.java



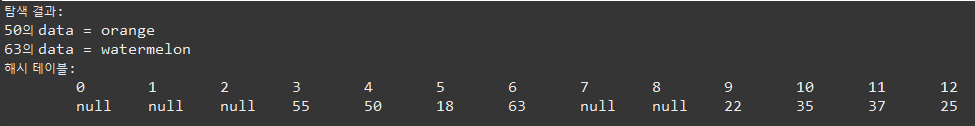
|  |
| --- |
| [Q4] 랜덤 조사 방식 [10점]  강의자료를 참고하여 랜덤조사방식 RandProbing 클래스를 구현하세요. 강의자료 p.35와 같이 테스트를 위한 main 메소드를 구현하고 실행하세요.  터미널 출력 화면을 캡처해서 본 문서에 첨부하고, 소스코드도 제출하세요. |

답변: RandProbing.java Q4main.java



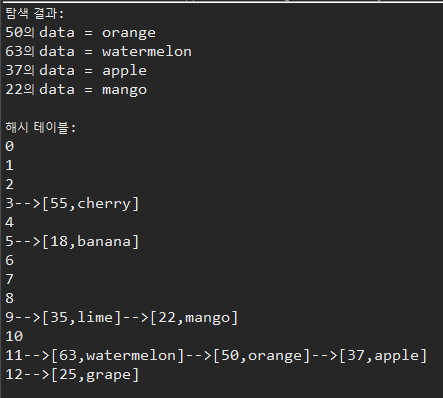
|  |
| --- |
| [Q5] 이중해싱 [10점]  강의자료를 참고하여 이중해싱 DoubleHashing 클래스를 구현하세요. 강의자료 p.40과 같이 테스트를 위한 main 메소드를 구현하고 실행하세요.  터미널 출력 화면을 캡처해서 본 문서에 첨부하고, 소스코드도 제출하세요. |

답변: DoubleHashing.java Q5main.java



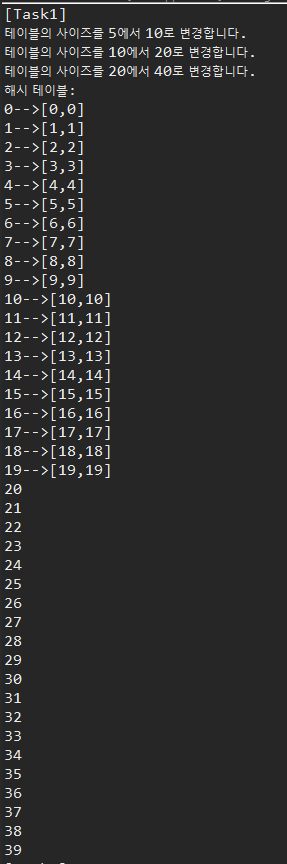
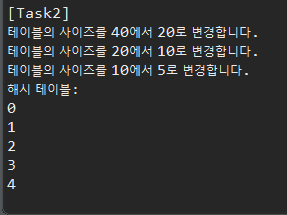
|  |
| --- |
| [Q6] 폐쇄주소방식 [10점]  강의자료를 참고하여 폐쇄주소방식 Chaining 클래스를 구현하세요. 강의자료 p.46과 같이 테스트를 위한 main 메소드를 구현하세요. Chaining 클래스의 print 메소드는 강의노트 p.46과 같이 출력해야 합니다 (‘해시 테이블:’ 로 시작하는 부분)  터미널 출력 화면을 캡처해서 본 문서에 첨부하고, 소스코드도 제출하세요. |

답변: Chaining.java Q6main.java



|  |
| --- |
| [Q7] 재해시 [20점]  Chaining 클래스에 재해시를 위한 private void rehash() 메소드를 구현하세요. get/put 메소드에서 적재율을 계산하고, 적재율이 0.75를 초과하거나 또는 0.25 밑으로 떨어지면(강의노트 p.49 참고) 해시 테이블을 2배로 늘리거나 1/2로 줄이도록 구현하세요. 처음에는 M=5로 설정하세요. 참고로, 해시테이블은 필요한 경우 계속 커지거나 작아질 수 있습니다. 하지만, 해시 테이블은 최소 M=5이며, 이보다 더 작아질 수는 없습니다. rehash 메소드는 호출될 때 마다 “테이블의 사이즈를 (A) 에서 (B)로 변경합니다.” 라는 메시지를 출력하도록 구현하세요. 이때, (A)는 현재 테이블 사이즈, (B)는 변경 후 테이블 사이즈입니다.  [Task 1]  아래와 같이 코딩하세요.  Chaining<Integer, Integer> ch = new Chaining<Integer, Integer>();  for(int i=0;i<20;i++) ch.put(i,i);  ch.print();  터미널 출력 화면을 캡처해서 본 문서에 첨부하세요.  Chaining 클래스에 public void delete(K key) 메소드를 구현하세요. 이 메소드는 인자로 주어진 key 값을 키 값으로 가지는 노드를 삭제합니다.  [Task 2]  아래와 같이 코딩하세요.  for(int i=0;i<20;i++) ch.delete(i,i);  ch.print();  터미널 출력 화면을 캡처해서 본 문서에 첨부하세요.  소스코드도 제출하세요. |

답변: Chaining.java Q7main.java

**끝! 수고하셨습니다 ☺**