알고리즘 실습 보고서

-Dijkstra algorithm-

전공 :컴퓨터공학과

분반 : 05반

학번 :201701988

이름 :김수빈

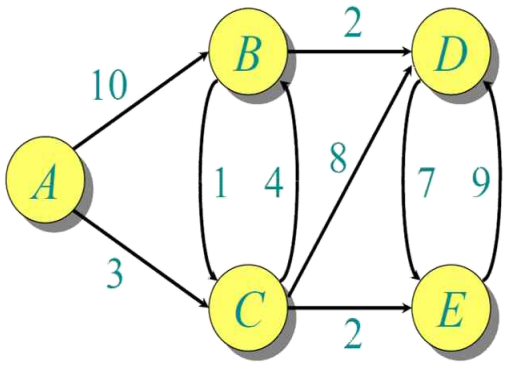
1. 실행 환경

본 실습은 Windows10 64bit, jdk 1.8.0_221, Eclipse EE가 설치된 환경에서 실행되었다

2. 과제 설명

출제된 과제에 대한 설명

Dijkstra algorithm을 사용하여 아래와 같은 그래프의 최단 경로 비용을 계산하는 프로그램을 구현하는 것이 이번 과제이다.



S 집합에 삽입하는 노드를 차례로 출력하고, S 집합에 노드를 삽입하고, 갱신된 d[v]의 값을 출력한다.

3. 문제 해결 방법

- 문제를 해결하기 위해 자신이 사용한 방법, 아이디어에 대한 설명

노드를 만들어 차례대로 ArrayList<Node> nodeList에 삽입하고, (노드 개수) x (노드 개수) 크기의 이차원 배열 path를 선언한다. path를 모두 Integer.MAXVALUE를 사용하여 초기화하고, 위 그래프의 edge값을 삽입한다. Dijkstra_algorithm 메소드를 사용하여 문제 해결을 진행한다.

Dijstra_algorithm 메소드는 그래프를 읽어 Dijstra algorithm을 사용하여 단계별 최소값인 점과 해당 점의 값을 출력하고 해당 점을 S집합에 삽입했을 때, 갱신되는 d[v]의 값을 출력하는 메소드이다.

먼저, 노드 개수에 해당하는 길이를 갖는 배열 d를 선언한다

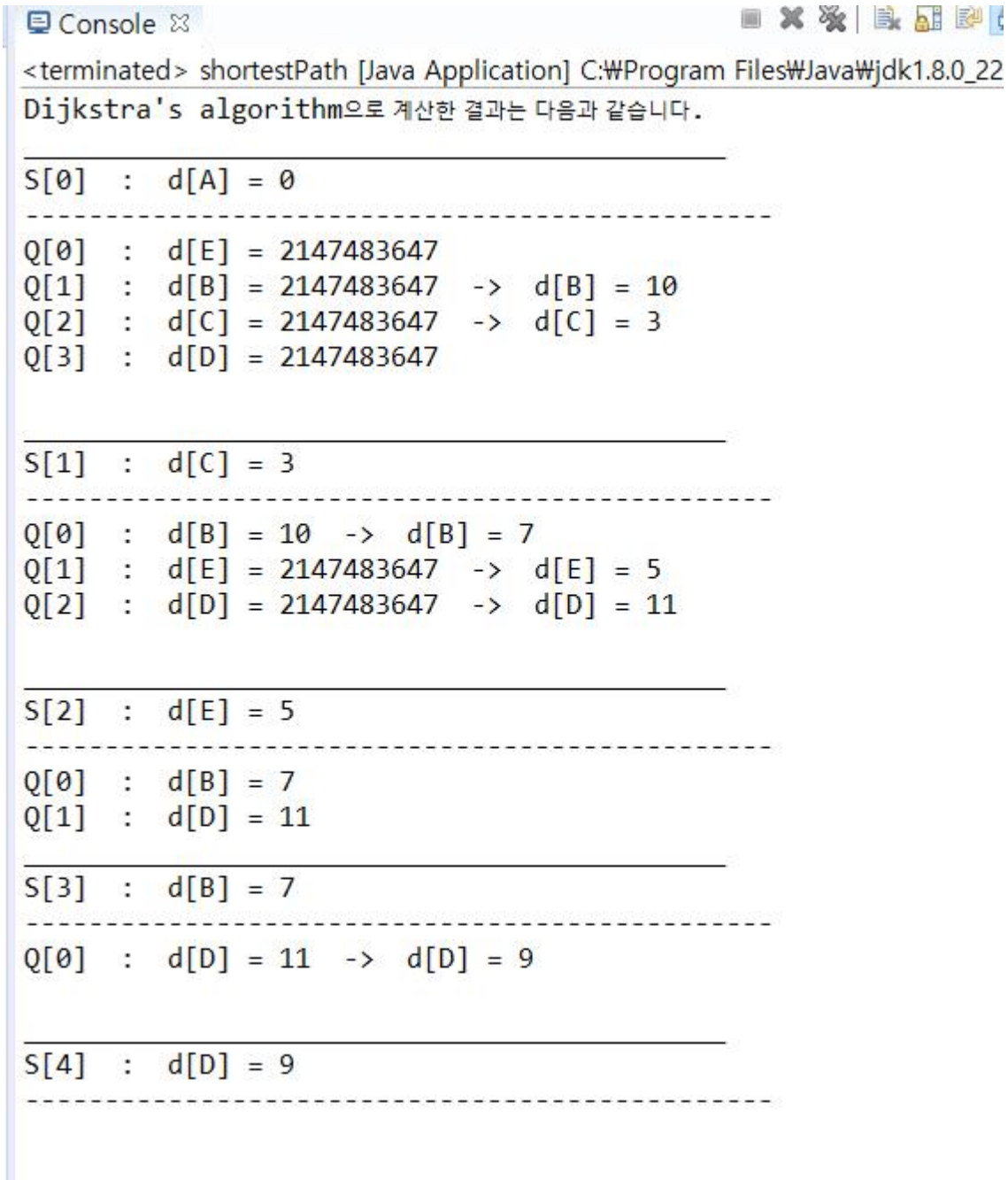
d 배열 내 값과 nodeList의 노드들의 key값을 Integr.MAXVALUE를 사용하여 초기화하고, d 배열에서 시작노드 s의 자리에는 0을 삽입하고, nodeList에서 시작노드 s에 해당하는 노드는 key값을 0으로 설정한다. nodeList를 복사해 priorityQueue 생성자의 매개변수로 전달하여 우선순위 큐 Queue를 생성한다. Queue가 Empty 상태가 될 때까지 다음을 반복한다.

Queue에서 최소값을 제거하고 제거한 최소값을 Node u에 저장한다. u를 집합 S에 삽입하고, Queue에 남아있는 노드들에 대해 다음을 반복한다. Queue 내 Node v에 대해, u와 v간 path가 Integer.MAXVALUE가 아닐 경우, d[u]+path[u][v]의 값이 d[v]보다 작을 경우, d[v]의 값을 Queue.decrease_key 메소드를 통해 갱신한다.

노드를 인덱스로 할 수 없기 때문에, 노드는 ArrayList에 담되, 노드의 인덱스 값은 HashMap<String, Integer> nodeIndex를 통해 저장했다. 노드의 이름값을 주면, ArrayList<Node> nodeList 내 인덱스 값을 반환한다. ArrayList 라이브러리 내 indexOf 메소드는 노드 객체를 통해 인덱스를 알아내는 것인데, 노드의 key값을 알아내기에 복잡함이 있을 것 같

아 HashMap에 따로 저장하여 구현했다.

결과 화면



5. 느낀점 및 고찰

지난 Heap 과제 때 구현했던 우선순위 큐를 사용해서 구현하여, 큰 어려움은 없었습니다.