**효율적인 택배 배달 알고리즘**

**요구사항 명세서**

**2013104042**

**강명훈**

**목차**

1. 시스템 개요 및 요약(목표)

2. 개발 환경 및 팀 구성

1) 개발 환경

2) 팀 구성

3. 외부 인터페이스와 자료 흐름

- 사용자 및 시스템 시각에서의 특성

- 입출력 양식(개념적 내용을 기반으로 가상 입/출력 표시)

- 자료흐름도, 자료사전 등

4. 기능 명세

- 필요기능 기술

5. 성능요구 사항

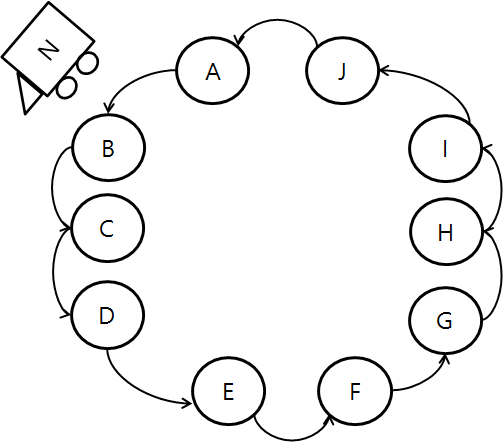
- 시간적 요구

- 효율성 요구

6. 예외조건 및 처리

7. 참고자료

**1.시스템 개요 및 요약(목표)**

1) 개요

어떠한 지역에 순환 고리의 배달 장소가 있다. 택배 회사에서 A ~ J 지역의 각 배달 장소에서 다른 장소로 물건을 배달해야 한다. 배달 차량의 공간은 N으로 한정이 된다.(N은 사용자로부터 입력을 받는다.) 각 배달 장소를 순차적으로 돌면서 배달한다. 이 때 배달은 A -> B -> C -> … -> I -> J -> A로 순환한다.

1. 각 지역에서 다른 지역으로 배달 해야 하는 배달물의 수

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 보관 장소 | | | | | | | | | | |
| 배달 할 장소 |  | A | B | C | D | E | F | G | H | I | J |
| A | 0 | 2 | 3 | 10 | 5 | 3 | 11 | 5 | 8 | 3 |
| B | 2 | 0 | 9 | 11 | 10 | 6 | 1 | 8 | 5 | 2 |
| C | 15 | 16 | 0 | 3 | 2 | 15 | 6 | 8 | 19 | 20 |
| D | 8 | 18 | 5 | 0 | 6 | 15 | 18 | 4 | 5 | 1 |
| E | 16 | 20 | 10 | 15 | 0 | 1 | 4 | 15 | 7 | 18 |
| F | 8 | 10 | 12 | 9 | 6 | 0 | 8 | 7 | 6 | 3 |
| G | 12 | 18 | 8 | 20 | 11 | 6 | 0 | 10 | 15 | 4 |
| H | 1 | 9 | 5 | 6 | 7 | 3 | 0 | 0 | 2 | 3 |
| I | 20 | 15 | 3 | 10 | 15 | 4 | 3 | 2 | 0 | 8 |
| J | 2 | 19 | 20 | 15 | 8 | 7 | 10 | 6 | 3 | 0 |

2) 목표

각 배달 장소를 순차적으로 돈다면 최소 순환으로 모든 물건을 배달 할 수 있는 배달 순서는 어떻게 되는가?

**2. 개발 환경 및 팀 구성**

1) 개발 환경

MS Visual Studio 2010 C++, C#을 통하여 구현(GUI 프로그래밍)

OS : Windows 8.1 K (64bit)

Processor : Intel® Core™ 4Quad i5-3230M CPU @ 2.60GHz

RAM : 8GB

2) 팀 구성

- 2인 구성 -

2013104042 컴퓨터공학과 강명훈

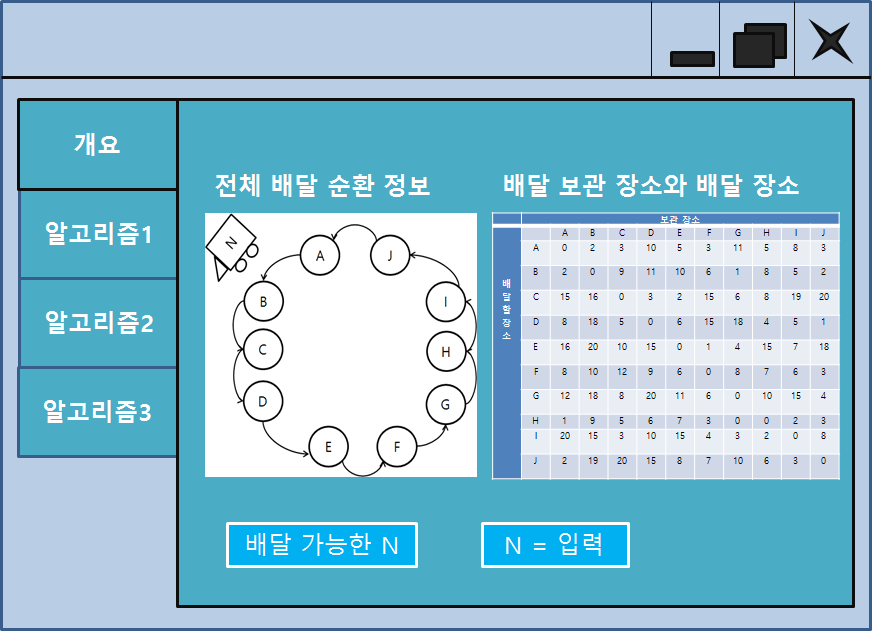
2013104087 컴퓨터공학과 신선경

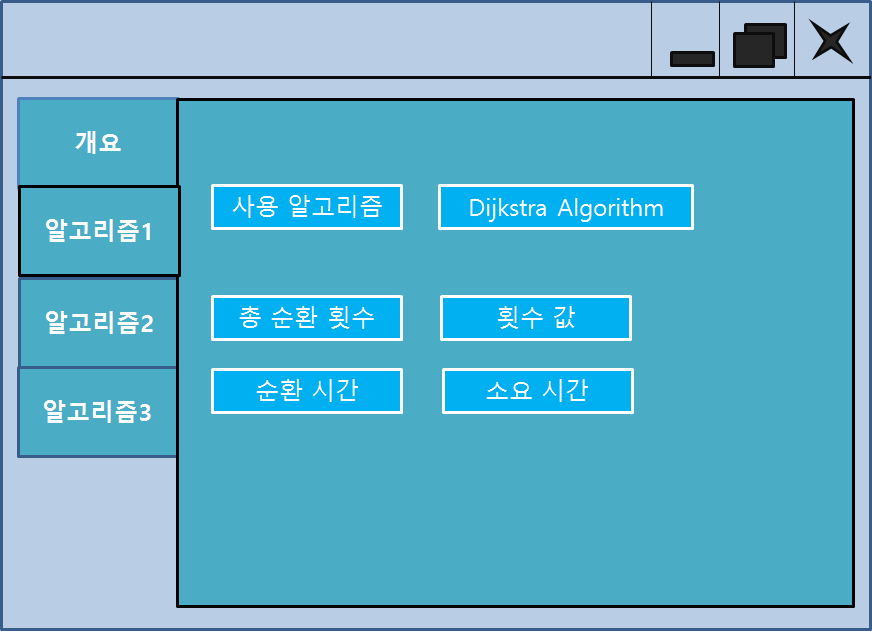
역할분담

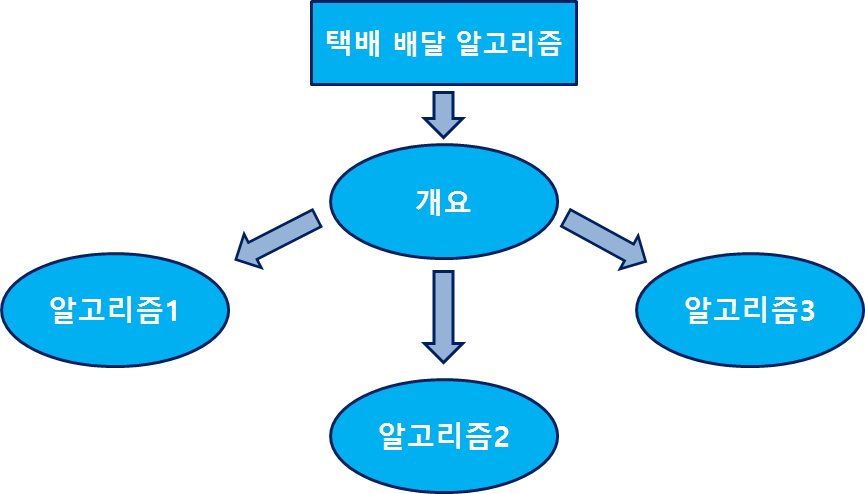
알고리즘 구상과 코딩은 같이 할 예정이며 C#으로 구현한 GUI 쪽은 공부 후 역할 분담 예정입니다.

**3. 외부 인터페이스와 자료 흐름**

- 사용자 인터페이스



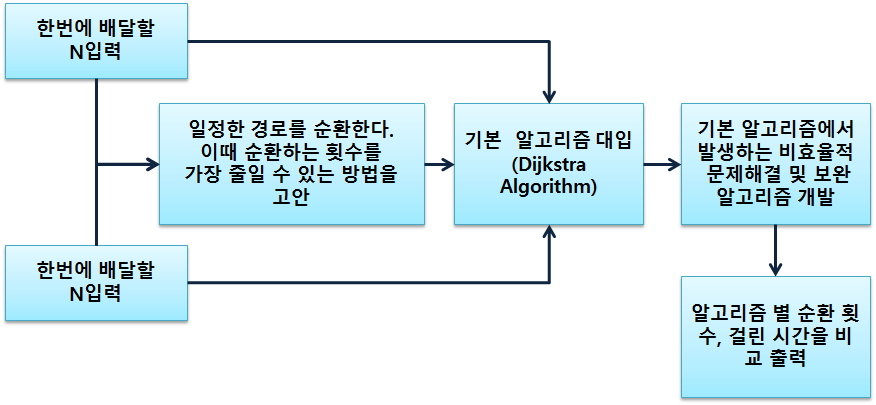


- 프로그램 동작원리

## 프로그램이 실행 되면 문제에 대한 정보를 가지고 있는 개요와 그러한 문제를 해결 하기 위해 사용된 알고리즘을 이용한 방식을 알아 볼 수 있는 창이 뜬다. 또한 문제에서 사용자가 필수적으로 주어야 할 N값 입력창도 뜨게 한다. 해당 알고리즘을 누르게 되면 그러한 알고리즘에서 배달하는데 필요한 순환 횟수 출력과 그러한 순환을 실행하는데 걸리는 시간을 출력 할 수 있게 한다.

- 입출력 양식(개념적 내용을 기반으로 가상 입/출력 표시)

사용자는 알고리즘 실행 시 출력되는 창에서 반드시 문제 해결에 필요한 임의의 N값을 입력해야 한다. 이러한 N값과 프로그램에 주어진 알고리즘을 기반으로 10개의 도시를 순환하는 횟수와 그에 따른 시간을 출력할 수 있게 한다.

- 자료흐름도, 자료사전 등

* 자료사전

순환하는 경로

* 이미 정해져 있다. (A -> B -> C -> … -> I -> J -> A)

배달 물품 목록

* 배달할 물건이 저장 되어 있는 장소 및 개수
* 배달할 물건이 배달 되어야 하는 장소

순환 횟수

* 한번의 알고리즘이 실행 되어 경로를 한번씩 돌 때 마다 값을 하나씩 증가시켜 최종적으로 나오는 결과 값을 받음

입력 값 N

* 사용자가 입력하는 값으로 택배가 한번에 옮길 수 있는 물건의 양을 정한다.

**4. 기능 명세**

필요 기능 기술

1. N을 사용자로부터 입력 받는 기능
2. 각 배달 지역마다 다른 지역으로 배달해야 할 물건을 입력 받는 기능
3. 지역을 순회 시 알고리즘에 따라 수행을 한 후 한번 순회를 할 때마다 순회의 횟수를 재는 기능
4. 알고리즘 수행 시에 순회 횟수뿐만이 아닌 순회 시에 걸리는 시간을 찾는 기능
5. 주어진 결과 값을 화면에 출력하는 기능
6. 알고리즘끼리 비교하여 결과를 알려주는 기능

**5. 성능요구 사항**

- 시간적 요구(반응시간, 처리시간, 시간당 처리율 등)

위 환경에서 10개의 배달 지역을 가진 도시에서 한번에 N개의 택배를 배달하는데 걸리는 시간을 최소화 할 수 있는 알고리즘을 찾는 것을 목표로 한다.

- 효율성 요구 (사용한 알고리즘 제시 및 제시한 알고리즘들 간의 효율 비교 등)

1) Dijkstra 알고리즘

- 실행 시간은 m개의 변과 n개의 꼭지점을 가진 그래프에서 연결리스트나 배열 구조로 구현하고 단순한 선형 탐색으로 구현했을 때 실행 시간은 O(n^2)이 된다. 만약 희소 그래프 즉 n^2보다 작은 개수의 변을 갖는 그래프에 대해서는 인접 리스트와 Binary heap 또는 Fibonacci heap으로 구현한 우선순위 큐를 이용해 구할 수도 있다. 이때 Binary heap을 사용하면 O((m+n)logn)시간이 되고 Fibonacci heap을 통해 O(m+nlogn) 시간 까지 개선 할 수 있다.

2) Dijkstra 알고리즘 추가 및 변형

- 기존의 Dijkstra 알고리즘을 그대로 적용하면 너무나도 많은 낭비가 발생하기 때문에 이를 보완하기 위한 예외처리와 알고리즘 변화가 필요하다. 따라서 이러한 예외처리 추가 및 변형을 진행한다.

3) Dijkstra 보완 알고리즘 최종 완성

- 개발 단계에서 생각해낸 모든 예외처리와 변형을 집어넣은 최종 알고리즘을 완성 시킨다.

**6. 예외조건 및 처리**

1) 입력 데이터

입력 받는 N값은 무조건 양수여야만 한다. 그래야 알고리즘이 실행이 가능하다.

2) 알고리즘

기존의 알고리즘을 사용하되 그대로의 방식이 아닌 문제에 적용시켜 알고리즘을 실행 할 수 있게 한다.

그 외에는 구체적으로 알고리즘을 실행하는 단계가 아니기 때문에 예외조건과 처리 설명이 완벽하지 않다.

**7. 참고자료**

박정호 공저, 『컴퓨터 알고리즘』, 상조사, 2007 -> 알고리즘 참고

<http://cjmyun.tripod.com/Knowledgebase/DFD.htm> -> 자료 흐름도 참고

<http://ko.wikipedia.org/wiki/%EB%8D%B0%EC%9D%B4%ED%81%AC%EC%8A%A4%ED%8A%B8%EB%9D%BC_%EC%95%8C%EA%B3%A0%EB%A6%AC%EC%A6%98> -> 알고리즘 정리