

방문 순서 문제를 읽고 스스로 코드를 작성해 보는 날입니다.

주어진 문제 설명을 이해하고 전체 코드를 구현하는 문제 입니다.

[1] 문제 제목 : 방문 순서

파일명 : LCommunication.py

- 먼저 스스로 구현하고 학습한 후 솔루션 파일을 참고하세요. 솔루션 파일명은 다음과 같습니다.

[1] 문제 제목 : 방문 순서

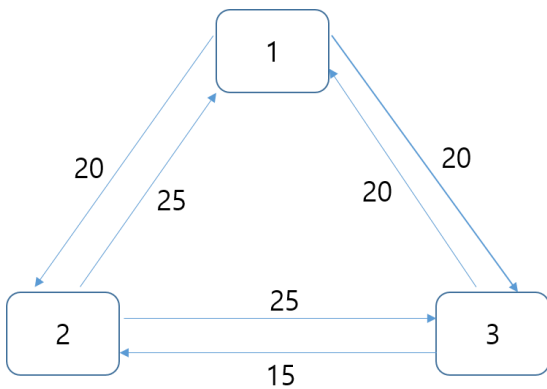
파일명 : LCommunication_solution.py

[1] 방문 순서

[문제설명]

L통신사 고객센터에서 인터넷 케이블 문제점을 접수 받은 고객 집을 방문하여 복구 작업을 하려고 합니다. 각 고객의 집과 집 사이를 이동하면서 소요되는 시간 정보를 주고 최소한의 시간이 소요될 수 있는 경로를 선택하여 작업을 시작하려고 합니다. 출발은 항상 1번 집에서 시작해야 합니다.

다음 예와 같이 방문해야 하는 집이 3가구이고 1번 -> 2번 -> 3번 -> 1번 순서로 전체 방문을 마치는 시간은 $20+25+20 = 65$ 분이고 1번->3번->2번 -> 1번 순서로 전체 방문을 마치는 시간은 $20+15+25 = 60$ 분으로 최소 시간은 60분을 소요하여 전체 서비스를 진행할 수 있습니다.



최소 비용이 소요되는 경로를 찾기 위해 다음과 같은 프로세스로 진행하려고 합니다.

- 각각의 집으로 이동할 수 있는 비용을 정리합니다.
- 1번 집부터 방문하고 한번 방문했던 집을 다시 방문하지 않도록 합니다.
- 모든 집을 다 수리 했다면 다시 출발지인 1번 위치로 돌아와야 합니다.

최소 비용이 소요되는 프로그램 구조는 다음과 같은 방법으로 구현하려고 한다.

- 한번 방문한 경로를 다시 방문하는 일을 막기 위해 방문 여부를 판단할 수 있는 구조를 만듭니다..
- 입력 정보에 출발지와 도착지 그리고 비용 정보를 이용하여 그래프로 상태를 표기 한다. 일반적으로 그래프의 상태는 이차원 리스트를 이용하여 작성합니다.

- 첫 출발지부터 시작하여 이미 방문한 곳은 제외하고 남아 있는 방문지를 찾아가며 그 때까지의 비용을 계산하려 합니다. 어떤 경로를 선택하는 것이 최적인지는 모든 경로를 비교해 봐야 하므로 현재 위치를 선택 했을 때와 선택하지 않았을 때의 비용을 비교하여 비용 절감되는 경로를 선택하도록 하며 이를 위해 재귀호출을 이용한다.

고객 수 `houses`, 각 가구들 사이의 소요 시간 정보인 `times` 가 `solution()`의 매개변수에 전달되며 반드시 모든 고객 집을 방문해야 합니다. 어떤 경로로 작업을 진행하는 것이 가장 최소의 시간을 들일 수 있는지 최소비용 계산 프로그램을 작성하려고 합니다. 위의 프로세스로 진행하기 위해 프로그램을 작성하였고 정상적으로 구동되어 결과를 얻기 위해 다음 코드의 빈칸에 `@@@` 로 표기된 부분을 채우세요.

[입력값 설명]

고객 집 수인 `houses`, 집과 집 사이에 소요되는 시간 정보 `times` 가 `solution()` 매개변수에 주어집니다.

- `houses` 는 3 이상 10 이하인 자연수입니다.
- `times` 에 주어진 정보의 크기는 `houses` 보다는 크고 `houses * (houses - 1)` 보다 작습니다.
- `times` 의 정보는 [시작번호, 도착지번호, 소요시간]의 형태로 구성됩니다.
- 한번 방문한 집은 다시 들리지 않습니다.
- 1 번 집에서 시작하여 모든 집을 방문하고 작업을 끝낼 수 있는 경로는 반드시 존재합니다.

[출력값 설명]

모든 집을 방문하여 서비스 할 수 있는 시간 중 가장 최소 시간(분으로 표기됨)를 return 합니다.

[입출력 예]

houses	times	return
3	[[1,2,20],[2,3,25],[2,1,25],[1,3,20],[3,2,15],[3,1,20]]	60