

뉴.진.스

다익스트라

2023-08-18(금)

# 목차

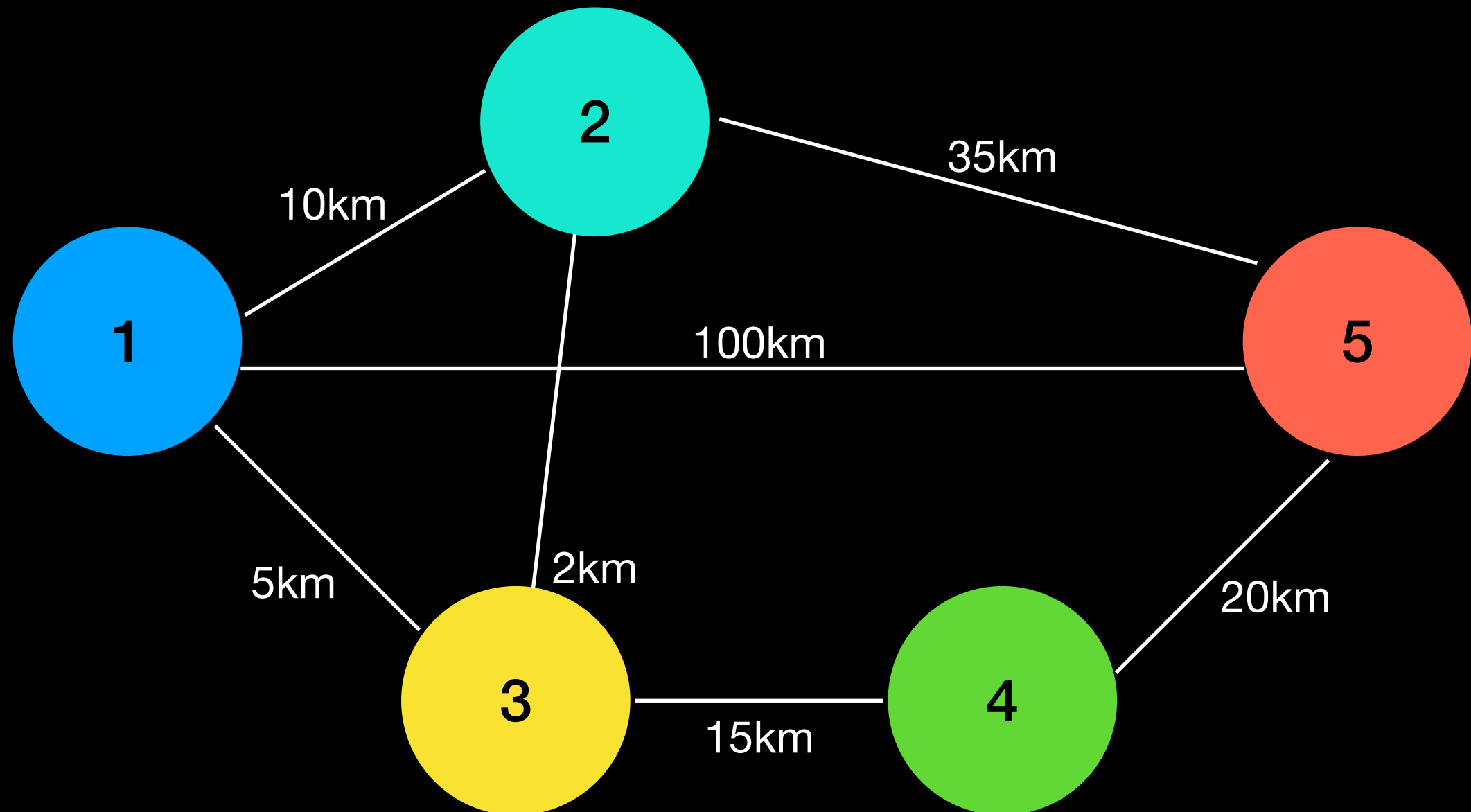
- 1.최단경로
- 2.다익스트라 개념
- 3.다익스트라 코드
- 4.Live Coding



최단경로

# 최단경로

- 특정 노드에서 다른 노드까지 갈 수 있는 최단경로를 찾는 알고리즘
- 대표적으로 다익스트라, 플로이드-워셜, 벨만-포드 알고리즘이 있음.



다익스트라 개념

# 다익스트라 개념

간선의 가중치가 없을 때 어떤 알고리즘으로 최단경로를 찾을 수 있었나요?

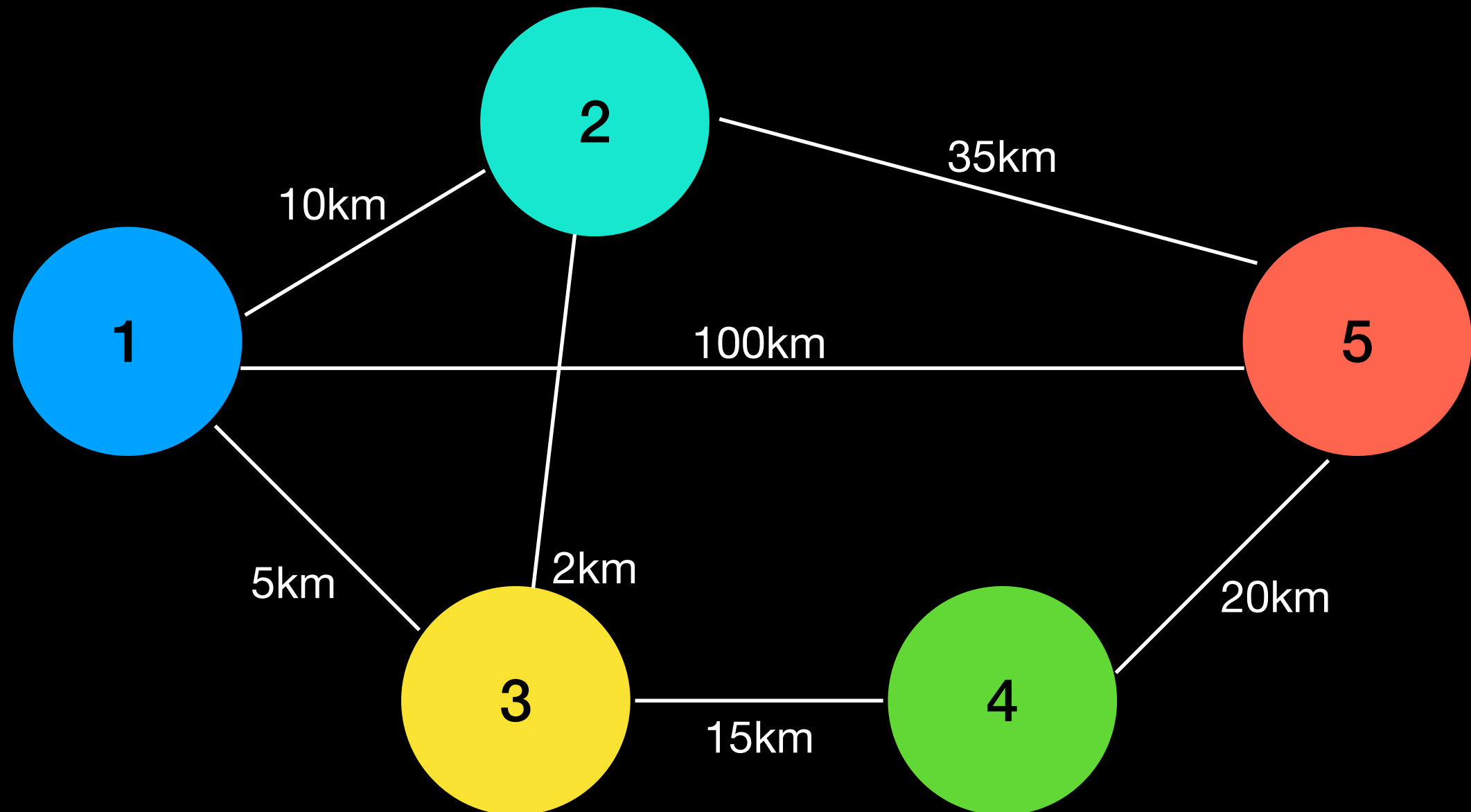
BFS 알고리즘을 구현하기 위해 필요한 자료구조는 무엇이었나요?

우리가 배우는 최단경로의 알고리즘에는 가중치가 있어요,  
여기에는 어떤 자료구조를 사용하는게 좋을까요?

힌트: (????) 큐

# 다익스트라 개념

- 우선순위 큐(이하 힙)를 통해서 특정노드로부터 간선이 이어진(간접적이든, 직접적이든) 모든 정점까지의 거리를 구하는 알고리즘.
- 힙을 통해서 더 짧은 거리의 간선이 있으면 그 값을 갱신하는 식으로 진행

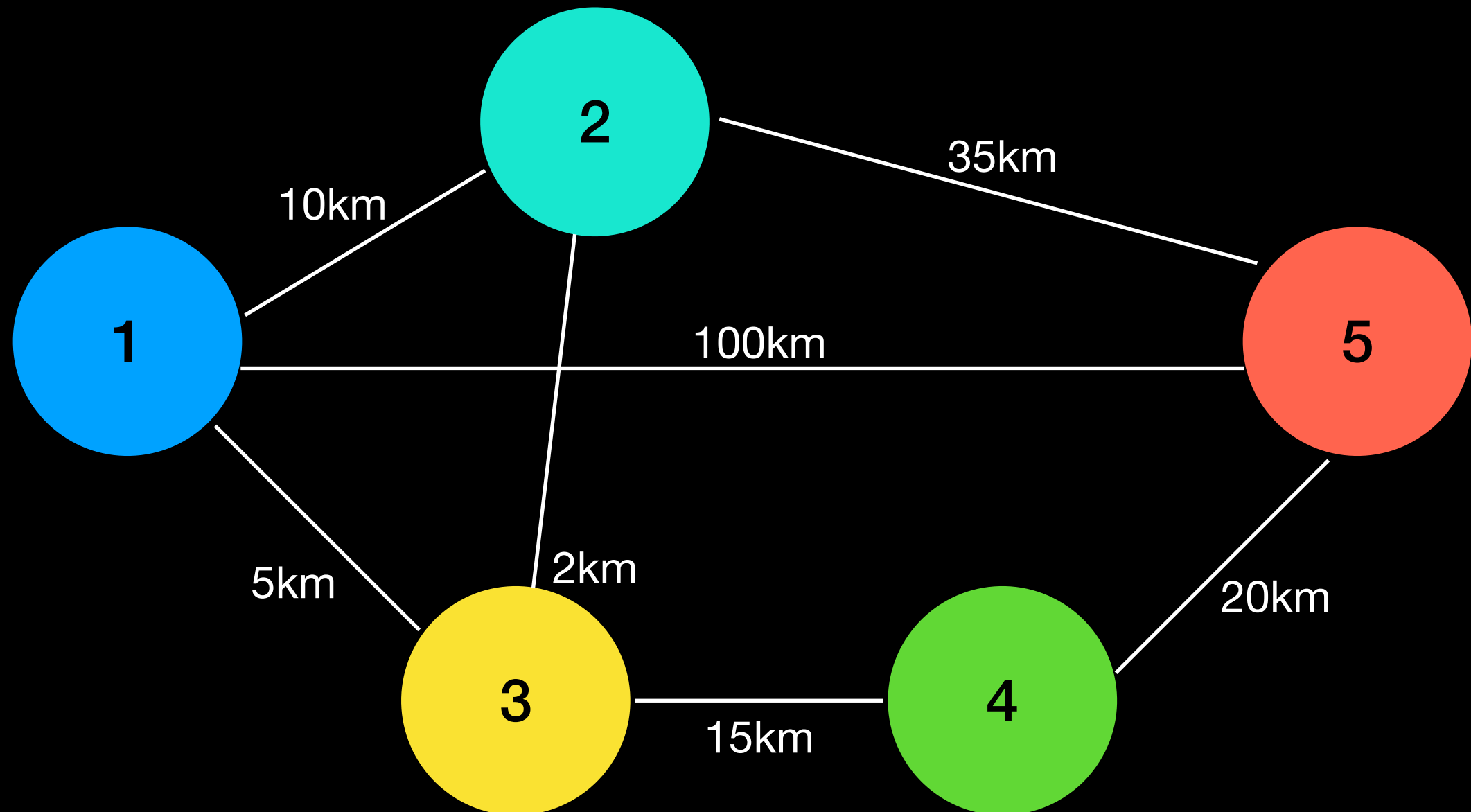


거리 저장 배열

|          |          |          |           |           |
|----------|----------|----------|-----------|-----------|
| (1)<br>0 | (2)<br>7 | (3)<br>5 | (4)<br>20 | (5)<br>40 |
|----------|----------|----------|-----------|-----------|

최소 힙

|           |           |           |  |  |
|-----------|-----------|-----------|--|--|
| (5)<br>40 | (2)<br>10 | (5)<br>42 |  |  |
|-----------|-----------|-----------|--|--|





다익스트라 코드

# 다익스트라 코드

1753 최단경로



# 다익스트라 코드

- 정점의 개수, 간선의 개수, 시작 정점 입력받기

```
let ve = readLine()!.split(separator: " ").map { Int($0)! }  
let (v, e) = (ve[0], ve[1])  
let startIndex = Int(readLine()!)! - 1
```

# 다익스트라 코드

- 준비물

힙, 가중치를 넣어놓을 배열, 간선 정보를 넣어놓을 배열  
그리고 시작점을 입력한다.

```
var weights = [Int](repeating: Int.max, count: v)
var edges = [[(vertex: Int, weight: Int)]](repeating: [], count: v)
var heap = Heap<(vertex: Int, weight: Int)> { $0.weight < $1.weight }
weights[startIndex] = 0
heap.enheap((vertex: startIndex, weight: 0))
```

# 다익스트라 코드

- 간선의 정보를 입력받는다.  
이때 양방향 간선이 아닌, 단방향 간선인 것 유의

```
for _ in 0..<e {  
    let uvw = readLine()!.split(separator: " ").map { Int($0)! - 1 }  
    let (u, v, w) = (uvw[0], uvw[1], uvw[2])  
    edges[u].append((vertex: v, weight: w + 1))  
}
```

# 다익스트라 코드

- 이제 힙에서 `deheap()`을 한다.  
`deheap()`의 결과값이 기존에 있는 가중치 보다 클 경우에는 스킵한다.

```
while !heap.isEmpty {  
    let info = heap.deheap()  
    if info.weight > weights[info.vertex] {  
        continue  
    }  
    // 밑의 코드는 다음 페이지에서 설명할 예정  
}
```

# 다익스트라 코드

- 이제 자신과 연결된 간선을 살펴본다.  
새로운 값이 기존의 값보다 작을 경우, 기존의 값을 갱신해주고  
힙에다가 해당 정점 번호와 새로운 값을 넣어준다.

```
while !heap.isEmpty {  
  // 위쪽 코드는 앞 페이지 참고  
  for edge in edges[info.vertex] {  
    let newWeight = info.weight + edge.weight  
    if newWeight < weights[edge.vertex] {  
      weights[edge.vertex] = newWeight  
      heap.enqueue((vertex: edge.vertex, weight: newWeight))  
    }  
  }  
}
```

# 다익스트라 코드

- 그럼 이제 출력한다.  
끝.

```
var result = ""
weights.forEach {
    result += "\($0 == Int.max ? "INF" : String($0))\n"
}
print(result)
```



프로이드  
[정신분석학]  
위대한

쉬는 시간

18223 민준이와 마산 그리고 건우



**8 주간 들어주셔서  
정말 감사합니다.**