최단 경로 – 플로이드 와셜 알고리즘

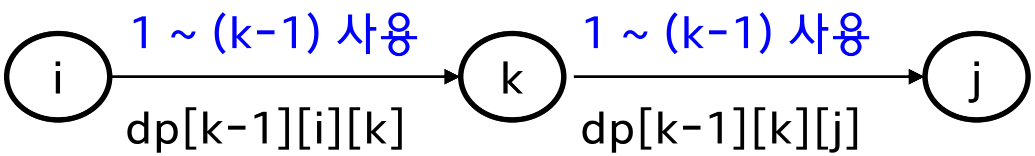
Shortestpath

- 모든 쌍의 최단 경로를 구하는 문제에 사용되는 알고리즘
- dp[k][i][j] = i ~ j로 이동하는 최단 경로, 이 때 중간에 방문할 수 있는 점은 1,2, ⋯ , k

  ① k가 경로에 없는 경우 : 1 ~ (k-1) 정점을 이용해서 i → j로 이동하는 최단 경로

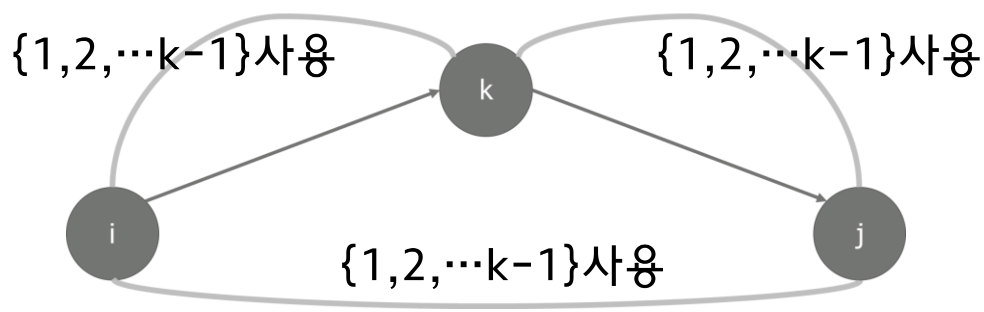
  ☞ dp[k-1][i][j]

  ② k가 경로에 있는 경우 : 1 ~ (k-1) 정점을 이용하여 i → k → j로 이동하는 최단 경로



  ☞ dp[k-1][i][j] + dp[k-1][k][j]

- dp[k][i][j] = min(dp[k-1][i][j], dp[k-1][i][k] + dp[k-1][k][j]) (k>=1)
- 초기값 : dp[k][i][j] = A[i][j] (k == 0)



최단 경로 – 플로이드 와셜 알고리즘

Shortestpath

- 구현할 때에는 2차원 배열로 구현하면 된다.

```
for(k = 1; k <=N; ++k){
    dp[k-1]을 이용하여 dp[k]를 구한다.
}
```

=> 아래 코드를 응용하여 플로이드 문제를 해결한다.

```
for(int k = 1; k <= N; ++k){
    for(int i = 1; i <= N; ++i){
        for(int j = 1; j <= N; ++j){
            dp[i][j] = min(dp[i][j], dp[i][k] + dp[k][j]);
        }
    }
}
```