Pervasive Elastic MetaLearning Laboratory
Department of Computer Engineering
Hongik University

HW13 실습

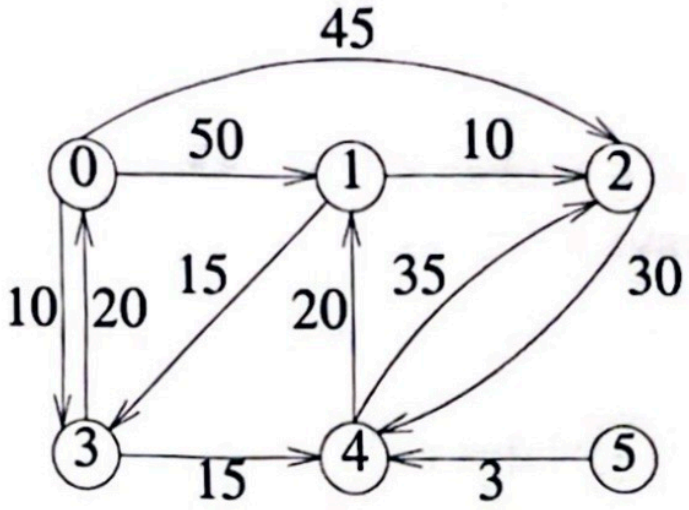
Shortest Path

최단 경로

- 단일 시점/모든 종점
  - 음이 아닌 가중치 : Dijkstra
  - 일반적 가중치 : BellmanFord
- 모든 쌍
  - Dijkstra 노드 개수만큼 반복?
  - Floyd-Warshall

단일 시점/모든 종점
음이 아닌 가중치

- 문제 조건 : $G = (V, E)$, length(i, j) > 0, 시발점 v
- 문제 : v와 G의 나머지 모든 노드들과의 최단거리
- → 다익스트라



(a) 그래프

경로	길이
1) 0, 3	10
2) 0, 3, 4	25
3) 0, 3, 4, 1	45
4) 0, 2	45

(b) 0에서부터의 최단 경로

```
void MatrixWdigraph::ShortestPath(const int n, const int v)
{// dist[j],0 ≤ j< n은 n개의 정점을 가진 방향 그래프 G에서 정점 v에서 정점 j
// 까지의 최단 경로 길이로 설정됨. 간선의 길이는 length[i][j]로 주어짐.
   for (int i = 0; i < n; i++) { s[i] = false; dist[i] = length[v][i]; } // 초기화
   s[v] = true;
   dist[v] = 0;
   for (i=0; i<n-2; i++) { // 정점 v로부터 n-1개 경로를 결정
      int u = Choose(n); // choose returns a value u such that:
                         // dist[u] = minimum dist[w], where s[w]=false
      s[u] = true;
      for (int w=0; w<n; w++)
         if(!s[w] && dist[u] + length[u][w] < dist[w])
            dist[w] = dist[u] + length[u][w];
    } // end of  for (i=0; ...)
}
```

단일 시점/모든 가중치
일반적인 경우

- 벨만-포드
  - 음의 가중치가 있는 그래프에 대해 최단 경로 계산 가능
  - 음의 사이클 여부 감지 가능

```
void MatrixWDigraph::BellmanFord(const int n, const int v)
{// Single source all destination shortest paths with negative edge lengths.
    for(int i=0; i<n; i++) dist[i] = length[v][i]; // dist 초기화


    for(int k=2; k<=n-1; k++)
      for(each u such that u != v and u has at least one incoming edge)
        for(each <i,u> in the graph)
          if(dist[u]> dist[i]+length [i][u]) dist[u] = dist[i] + length [i][u];
}
```

최단 경로를 계산하는 Bellman과 Ford 알고리즘

모든 쌍 최단경로

- 다익스트라를 노드 개수만큼 반복?
- 플로이드
- 2차원 경로 배열

$$A^k[i][j] = \min\{A^{k-1}[i][j], A^{k-1}[i][k] + A^{k-1}[k][j], k \geq 0$$
$$A^{-1}[i][j] = length[i][j]$$

```
void MatrixWDigraph::AllLengths(const int n)
 { // length[n][n]은 n개의 정점을 가진 그래프의 인접 행렬이다.
  // a[i][j]는 i와 j 사이의 최단 경로의 길이이다.
    for(int i=0; i<n; i++)
       for(int j=0; j<n; j++)
          a[i][j] = length[i][j]; // length를 a에 복사
    for(int k=0; k<n; k++)        // k이하 번까지의 정점들을 고려한 경로에 대해
       for(i=0; i<n; i++)         // 가능한 모든 정점의 쌍에 대해
          for(int j=0; j<n; j++)
             if((a[i][k]+a[k][j]) < a[i][j]) a[i][j] = a[i][k] + a[k][j];
 }
```

모든 쌍의 최단경로

질문

- pemds81718@gmail.com
- 간단한 구글링으로 알 수 있는 내용은 답변하지 않습니다.