|  |
| --- |
| **컴퓨터네트워크** |
| **과제 #04 문제 및 보고서** |

|  |  |
| --- | --- |
| **이름** | 이풍헌 |
| **학번** | 201924548 |
| **소속**  **학과/대학** | 정보컴퓨터공학과 |
| **분반** | 01 (담당교수: 김태운) |

## <주의사항>

* 각각의 문제 바로 아래에 답을 작성 후 제출해 주세요.
* PLATO 제출 데드라인: 10/31(화)~11/13(월) 23:55까지
* 데드라인을 지나서 제출하면 0점
* 주말/휴일/학교행사 등으로 인한 데드라인 연장 없음
* 부정행위 적발 시, 원본(보여준 사람)과 복사본(베낀 사람) 모두 0점 처리함
* PLATO에 아래의 파일을 제출해 주세요
* 보고서
  + 보고서 파일명에 이름과 학번을 입력 해 주세요.
  + 보고서 첫 페이지(직전 페이지)에 이름, 학번, 소속을 작성해 주세요.
* 이번 과제를 수행하면서 소스코드, 스크립트, Makefile 등을 작성한 경우, 모든 파일을 압축하여 제출하세요(미제출 시 감점)

## <개요>

이번 과제는 Dijkstra 최단 경로 알고리즘을 구현하는 내용으로 구성되어 있습니다.

## <실습 과제>

# <Part I> 최단 경로 계산하기

|  |
| --- |
| **[Q 1] 최단 경로 계산하기 [배점: 20]**  강의자료에서 “Dijkstra’s algorithm” 내용을 참고하여, 아래의 네트워크(=그래프)에 대해 Dijkstra 알고리즘을 적용 하시오. (SRC: 출발지 노드)  \* 이번 문제는, DiAlgo를 구현하는게 아니고 종이와 펜으로 DiAlgo를 적용해 보는 문제입니다.    문제 1) 위 그래프에 대해, 강의자료 [참고: Dijkstra’s algorithm]에 있는 [Table 1, p.50]과 같은 테이블을 완성 하시오([Table 1] DiAlgo 단계별 계산 결과 테이블).  문제 2) 위 그래프에 대해, 강의자료 [참고: Dijkstra’s algorithm]에 있는 [Table 2, p.50]와 같은 테이블을 완성 하시오([Table 2] 최종 라우팅 테이블). |

답변 1)

답변 2)

텍스트, 스크린샷, 디자인이(가) 표시된 사진

자동 생성된 설명

# <Part II> 최단 경로 구현하기 (C 언어)

Dijkstra 최단 경로 알고리즘을 구현해 보겠습니다. 참고로, 아래의 구현 과정에서 사용하는 표기법 및 알고리즘은 강의자료에 있는 Dijkstra’s algorithm과 약간 다를 수 있습니다. 이번 과제에서 구현하게 될 Dijkstra 최단 경로 알고리즘은 아래와 같습니다(‘🡨’기호는 우측의 값을 좌측에 할당하는 연산을 의미합니다). 아래의 [최단 경로 알고리즘]에서 **하늘색**으로 표시된 부분을 하나씩 구현하는 과제입니다. 참고로, 네트워크는 그래프(G)로 표현할 수 있고, 그래프G는 정점(= 노드, 네트워크 호스트)과 간선(= 링크, 두 호스트 간 직접 연결)의 집합으로 구성됩니다.

|  |
| --- |
| 입력: 전체 네트워크에 대한 연결 상태 및 비용을 저장한 그래프 G (단, 비용 )  G에는 int n(정점의 개수)과 int[][] cost(간선의 비용, 거리)를 저장  출발지 노드의 인덱스 int v  출력: int distance[n] 배열  참고. distance[u]: 출발지 v로부터 u 까지 최단 경로의 **총** 경로 비용 |
| **shortest\_path**(G, v): // 출발지가 v인 경우, 모든 노드로의 최소 비용 경로 계산    For: 각 정점  distance[w] 🡨 cost[v][w]; // 초기화  While: 모든 정점이 집합 S에 포함될 때 까지  **print\_status** // 아래에 설명...  집합S’에 속하지 않는 정점 중, 최소 distance를가지는 정점 u를 **choose**  // u를 집합 S’에 추가  For: u에 인접하고 S’에 없는 각 정점 z 에 대해서  If: distance[u] + cost[u][z] < distance[z]  distance[z] 🡨 distance[u] + cost[u][z] |
| [Algo. 1] Dijkstra 최단 경로 알고리즘 |

G는 네트워크 그래프를 표현한 구조체이며, 2차원 배열(=매트릭스) 형태로 그래프 구조 및 각 간선에 할당된 비용(=거리)를 표현합니다. 그래프에서 두 노드가 인접(=직접 연결)한 경우에만 0 또는 양수로 표현된 cost(비용)값을 매트릭스 cost[x][y] 위치에 저장합니다. 예를 들어 0번과 3번 노드가 직접 연결(=인접)되어 있고, 둘 간의 비용(=거리)이 3인 경우, G의 cost[0][3]=3 입니다. 만약, 두 개의 노드가 인접하지 않다면, 둘 간의 비용(=거리)는 무한대(INF)로 표현합니다. 그리고, 자기 자신으로의 거리(비용)는 0입니다 (즉, cost[i][i] = 0).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **[Q 2] Dijkstra 최단 경로 알고리즘 구현 1: 구조체 표현 [배점: 10]**  그래프를 저장하기 위한 구조체 만들기: 아래의 구조체는 1) 네트워크에 존재하는 정점/호스트의 수를 나타내는 n 과, 2) 직접 이웃한 (= 인접한) 두 정점 간의 거리 비용을 저장하고 있는 cost 매트릭스로 구성되어 있습니다. 여기서, cost 매트릭스에는 직접 이웃한 정점 간의 거리를 0또는 양의 정수로, 나머지는 INF로 거리(비용)를 표현합니다.    구조체를 활용해서 네트워크 표현하기: 아래의 [그림 1.1] 그래프를 GraphType 구조체로 표현하는 방법은 [그림 1.2]와 같습니다.   |  |  | | --- | --- | |  |  | | 그림 1.1 | 그림 1.2 |   문제) 아래의 그래프를 GraphType 구조체로 코딩하세요. [그림 1.2]와 같이 코드를 캡처하고, 아래에 첨부 하시오. 참고로, INF는 매우 큰 정수로 설정된 매크로 상수 입니다. |

답변)

|  |
| --- |
| **[Q 3] Dijkstra 최단 경로 알고리즘 구현 2: print\_status [배점: 10]**  최단 경로 프로그램은 아래와 같이 두 개의 배열을 전역 변수로 선언하여 사용합니다.    ‘distance’ 배열은 출발 정점을 기준으로 각 정점까지 도달하는데 소요되는 거리 비용을 저장하는 1차원 배열 입니다. 지금까지 알아낸 최단 경로를 기준으로, 각 노드로 가는 최단 경로의 총 비용(거리)를 저장합니다. ‘found’는 출발 정점을 기준으로 최소 거리 계산이 완료된 정점을 표시하기 위한 배열이며, 예를 들어 found[u]가 1이라면 출발지로부터 정점 u까지의 최소 거리 계산이 완료되었다는 의미입니다 (= 집합에 가 포함되어 있다는 의미 입니다). ‘found’ 배열에는 TRUE 또는 FALSE가 저장되는데, 다음과 같이 매크로 변수를 선언된 상수입니다: #define TRUE 1 그리고 #define FALSE 0  프로그램 실행 시, 아래와 같이 각 STEP별 결과를 터미널에 출력하는 void print\_status (GraphType\* g, int step) {...} 함수를 구현 하시오. INF는 \*로 출력하세요. STEP이란, [Algo. 1]의 while 반복문에서 한번의 iteration을 의미합니다). 출발 노드가 1번 노드인 경우 출력은 아래와 같습니다. 출발노드를 1번 또는 본인이 원하는 번호로 설정하세요.    문제) print\_status 함수의 코드를 캡처하여 아래에 첨부하세요 |

답변)

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

|  |
| --- |
| **[Q 4] Dijkstra 최단 경로 알고리즘 구현 3: choose [배점: 10]**  [Algo. 1]에서 while 문의 매 Iteration 마다, 최단 경로 계산이 완료되지 않은 정점 중에서 최소의 거리 비용을 가진 정점을 선택해야 합니다. 이를 위해 int choose(int distance[], int n, int found[]) {...}를 구현하세요. 이 함수는 정점의 인덱스(정수 값)를 리턴합니다.  문제) choose 함수의 코드를 캡처하여 아래에 첨부하세요. |

답변)

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

|  |
| --- |
| **[Q 5] Dijkstra 최단 경로 알고리즘 구현 4: shortest\_path [배점: 10]**  최단 경로 계산 프로그램의 main 함수는 다음과 같이 구성되어 있습니다.    여기서 void shortest\_path(GraphType\* g, int start\_node) {...} 함수는 [Algo. 1] 을 구현한 함수입니다. 함수 호출 시, 네트워크 구성 정보가 저장된 GraphType 구조체 포인터, 그리고 시작 정점의 (정수) 인덱스 값을 인자로 전달합니다. 위의 [Algo. 1]을 참고하여 shortest\_path 함수를 구현하세요.  문제) shortest\_path 함수 코드를 캡처하여 아래에 첨부하세요. |

답변)

텍스트, 스크린샷, 디스플레이이(가) 표시된 사진

자동 생성된 설명

|  |
| --- |
| **[Q 6] Dijkstra 최단 경로 알고리즘 테스트 1 [배점: 20]**  아래의 그래프로 최단 경로 계산 프로그램을 실행하고, 터미널 출력을 캡처하여 아래에 첨부하세요.  출발 노드는 1번 또는 본인이 원하는 번호로 설정하세요. |

답변) 시작 0번

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

|  |
| --- |
| **[Q 7] Dijkstra 최단 경로 알고리즘 테스트 2 [배점: 20]**  아래의 그래프로 최단 경로 계산 프로그램을 실행하고, 터미널 출력을 캡처하여 아래에 첨부하세요.  출발 노드는 1번 또는 본인이 원하는 번호로 설정하세요. |

답변)

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

\*\* 전체 소스코드를 압축하여 첨부파일로 제출하세요.

**끝! 수고하셨습니다 ☺**