AI알고리즘

네트워크 모형(최단경로, CPM/PERT 등)

수퍼스타를 키우는 곳

전주대학교

최대흐름문제와 최소비용 용량제약 네트워크 문제

최대흐름 문제

최대흐름 문제

❖ 최대흐름 문제란

- 시작 마디에서 종료 마디까지 최대한 보낼 수 있는 용량을 찾는 문제
- 세계 2차 대전 당시 미국의 군수물자 이송 경로 중 가장 많이 보낼 수 있는 경로 찾기 위해 시작
  - 경로상 흐를 수 있는 요량이 제한됨

❖ 최대흐름 현실 문제

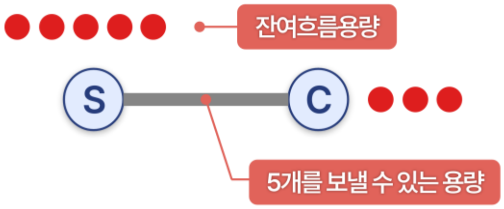
- 최대로 물을 많이 보낼 수 있는 방법은 무엇일까?
  - 댐 → 펌프장 → 상수도 처리장

❖ 최대흐름 문제 해결 순서

1. 시작 마디부터 종료 마디 사이에서 양의 흐름을 갖는 경로 우선 찾기(기준경로)
2. 기준경로를 구성하는 각 호들의 잔여흐름용량 중 최소흐름량이 그 경로의 최대흐름용량이 됨
3. 그러한 흐름용량들을 더하여 주어진 네트워크의 총 흐름량 계산

❖ 기호 정의

- C_{ij}^{R} = 마디 i에서 마디 j로의 $(i \rightarrow j)$ 사용가능한 잔여흐름용량
- C_{ji}^{R} = 마디 j에서 마디 i로의 $(j \rightarrow i)$ 사용가능한 잔여흐름용량



최대흐름 문제

❖ 최대흐름 문제 알고리즘

- 여러 경로 중 기준경로 찾기
  → 해당 마디들의 최소용량을 계산하여 그만큼만 보내기

- $[f_j, i]$
  - 물리적 모형
  - i 마디로부터 마디 j에 흐름이 발생
  - f_j는 마디 i 로부터 마디 j 까지의 흐름량

- 1단계
  - 모든 호(i,j)에 대해, 잔여흐름용량을 초기흐름용량으로 놓고 $f_j = \infty$, 시작마디 1에 라벨 $[\infty, -]$를 붙임

- 2단계
  - 마디 i 로부터 양의 잔여흐름용량을 가지고 있으면서 마디 j 로 직접 연결되는 라벨이 붙지 않은 마디들의 집합을 찾고, 이를 UN_i로 설정
  - (즉, 모든 $j \in UN_i$ 에 대해 $C_{ij}^R > 0$) 만일 $UN_i \neq \emptyset$ 라면 [단계3]으로 감

- 3단계
  - $C_{ik}^R = \max\{C_{ij}^R, j \in UN_i\}$을 만족하는 마디 $k \in UN_i$ 를 찾은 후 $f_k = C_{ik}^R$ 로 놓고 마디에 라벨 $[f_k, i]$를 붙임
    만일 $k = n$이면 종료마디까지 라벨이 붙여졌으므로 기준경로 탐색완료 [단계5]로 가거나 $i = k$로 놓고 [단계2]로 이동

- 4단계
  - 만일 $i = 1$이면 더 이상의 기준경로는 존재하지 않으므로 [단계 6]으로 가거나, 현행 마디 I 직전에 라벨이 붙여진 마디 b를 선택하고 현 반복작업에서는 마디 i를 고려 대상에서 제거한 후 $i = b$로 놓고 [단계 2]로 돌아감

- 5단계: 수정 네트워크의 결정
  - $N_m = (1, n_1, n_2, \cdots, n)$을 시작 마디 1부터 종료 마디 n까지의 m번째 기준경로를 구성하는 마디들의 순서를 나타내는 집합이라 정의
  - 그 기준경로에 대응되는 최대흐름량을
    $F_m = \min(f_1, f_{n_1}, f_{n_2}, \cdots, f_n)$로 계산
  - 기준경로를 따르는 각 호들의 잔여흐름용량 다음과 같이 수정
    - 흐름방향 $i \rightarrow j$에 대해, $(C_{ij}^R - F_m, C_{ji}^R + F_m)$
    - 흐름방향 $j \rightarrow i$에 대해, $(C_{ij}^R + F_m, C_{ji}^R - F_m)$

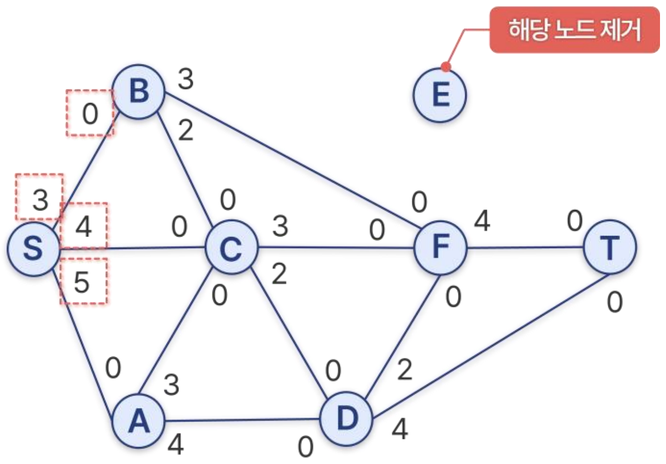
최대흐름 문제

❖ 최대흐름 문제 알고리즘

1. 출발지에서 목적지의 경로 중 하나의 경로 선택
2. 해당 경로의 미니멈 용량 찾아서 그만큼만 보내주기
3. 보낸 용량만큼 네트워크 수정
4. 끝날 때까지 1번부터 다시 반복

- 6단계: 해의 탐색
  - l개의 기준경로가 탐색되었다면 $F_m = F_1 + \cdots + F_l$로 최대흐름량 결정
  - 호(i,j)에 대한 초기 및 최종 잔여흐름용량을 각각 (C_{ij}^0, C_{ji}^0)과 (C_{ij}^R, C_{ji}^R)로 정의한 후 호(i,j)에 대한 최적흐름량을 $(x_{ij}^*, x_{ji}^*) = (C_{ij}^0 - C_{ij}^R, C_{ji}^0 - C_{ji}^R)$로 계산
  - 여기서, (x_{ij}^*, x_{ji}^*)는 각각 흐름 방향에 대한 최적흐름량

❖ 예제 소개



최대흐름 문제

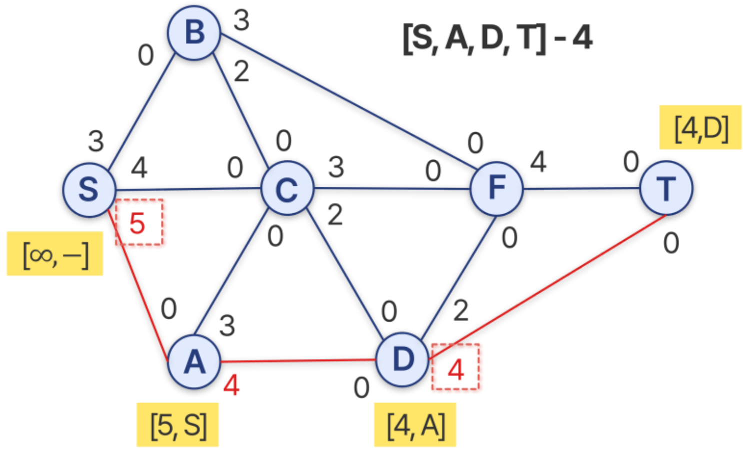
❖ 예제 알고리즘

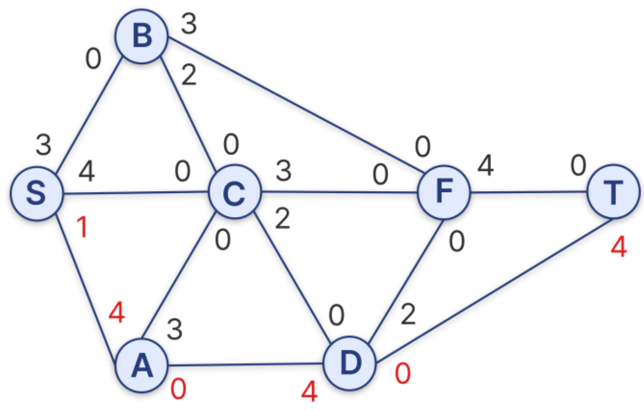
1. $f_S = \infty$, 시작마디 S에 라벨 $[\infty, -]$, $i = S$, [단계2]로 감
2. 마디 S로부터 직접 연결되는 마디들 중 양의 잔여용량흐름을 갖는 마디들의 집합 $UN_S \in \{A, B, C\}$를 결정, $UN_S \neq \emptyset$이므로 [단계3]으로 감
3. $\max\{C_{SA}^R, C_{SB}^R, C_{SC}^R\} = \max\{5, 3, 4\} = 5$를 제공하는 집합 내의 마디는 k=A이며 $f_A = 5$로 놓고 마디 A에 라벨 [5, S]을 붙임, $i = A$로 놓고 [단계2]로 감
6. $\max\{C_{DF}^R, C_{DT}^R\} = \max\{2, 4\} = 4$를 제공하는 집합 내의 마디는 $k = T$이며 $f_T = 4$로 놓고 마디 T에 라벨 [4, D]을 붙이고 $k = T$이므로 다음 단계로 감
7. $N_1 = \{S, A, D, T\}, F_1 = min\{\infty, 5, 4, 4\} = 4$, 잔여용량 수정

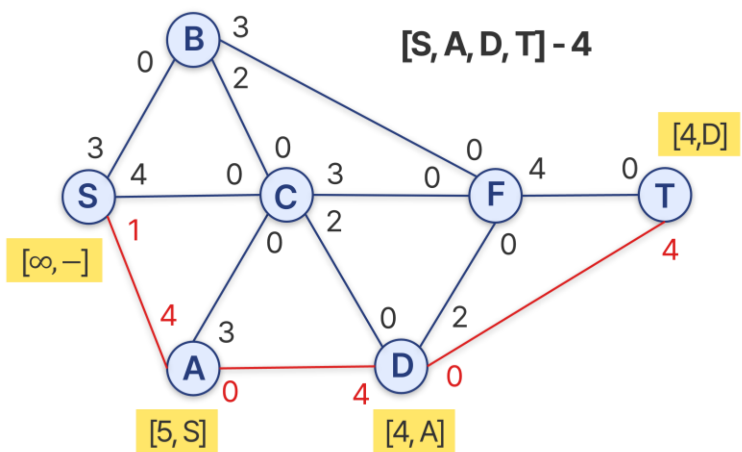
최대흐름 문제

❖ 예제

- 반복1



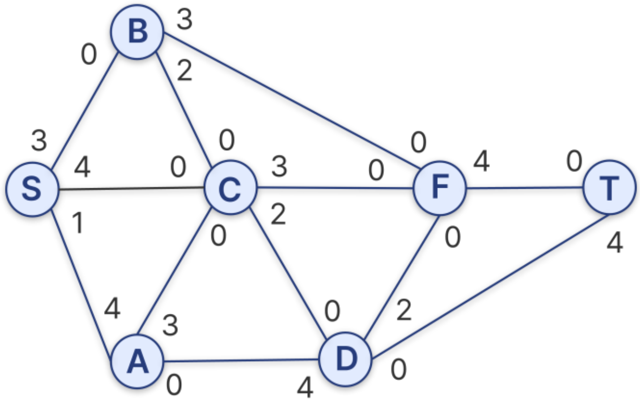


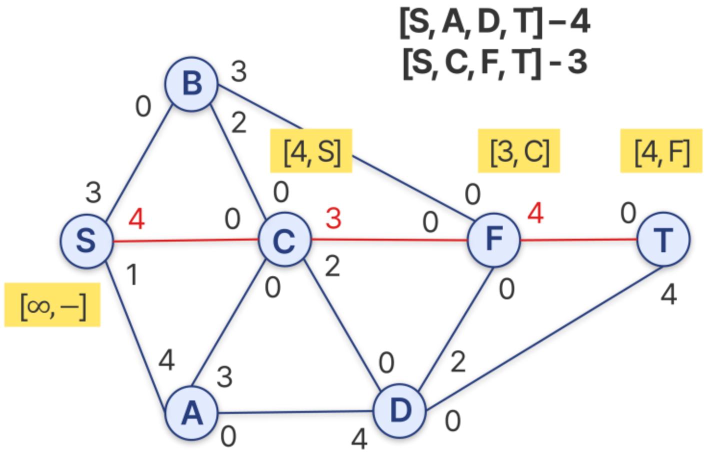


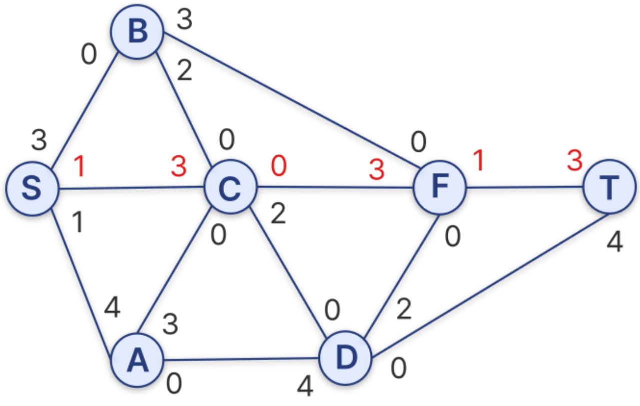
최대흐름 문제

❖ 예제

- 반복2



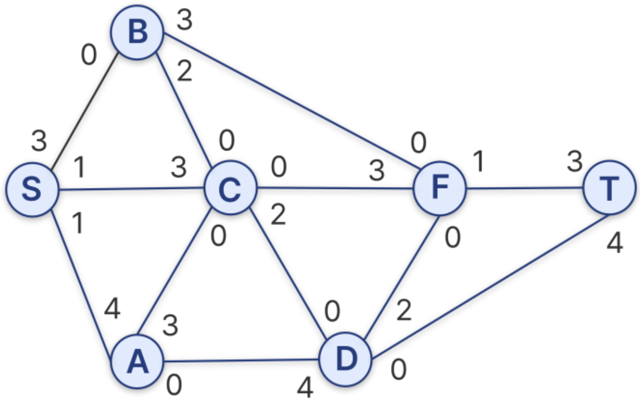




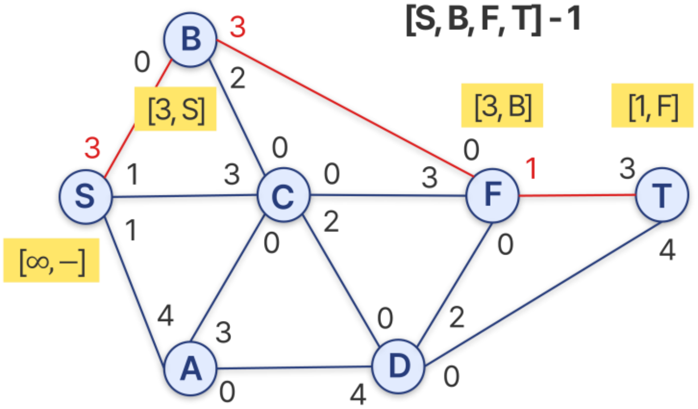
최대흐름 문제

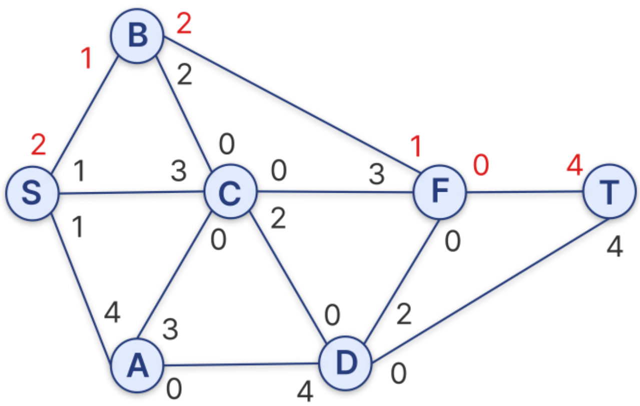
❖ 예제

- 반복3



[S, A, D, T] – 4
[S, C, F, T] – 3
[S, B, F, T] - 1

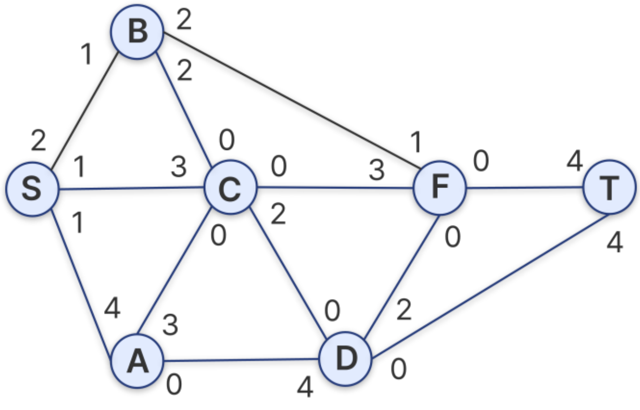


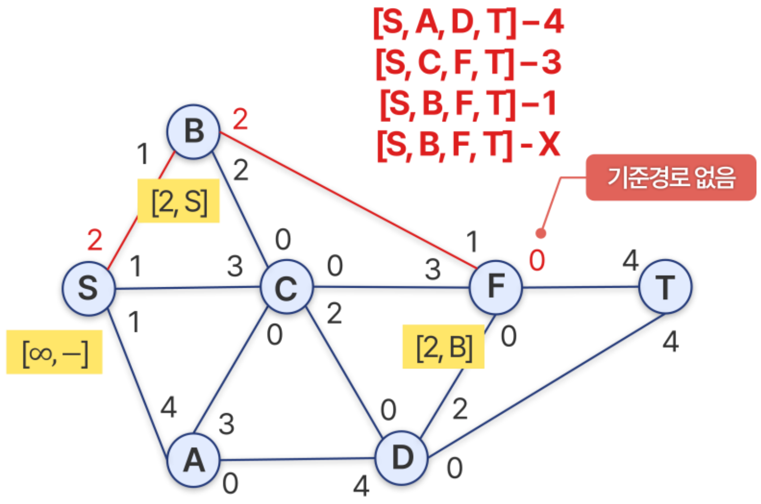


최대흐름 문제

❖ 예제

- 반복4



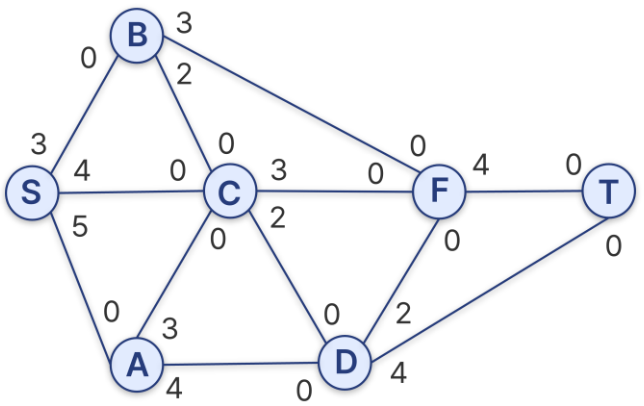


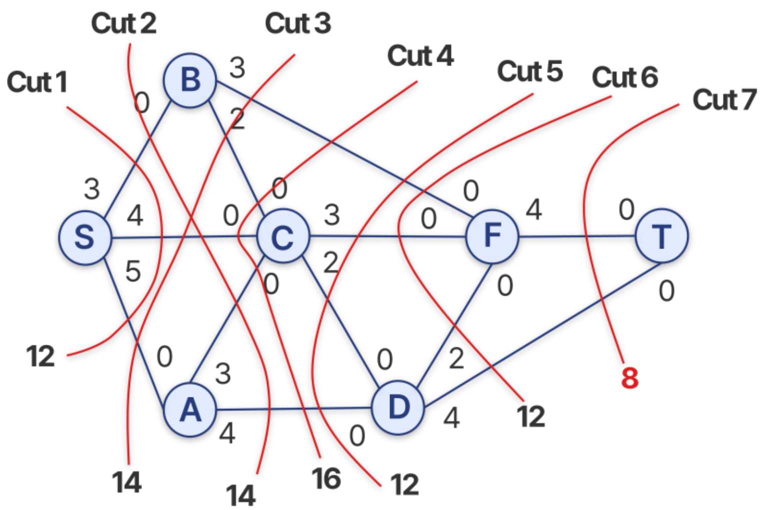
- 경로찾기 → 해당 용량만큼 보내기 → 네트워크값 수정 → 값이 없을 때까지 반복

최대흐름 문제

❖ Cut을 이용한 계산

- Cut을 이용한 최대흐름량 계산
  - 네트워크를 이용하여 가능한 Cut들을 열거하고 각 Cut들에 대한 Cut 용량 계산





❖ 선형계획모형을 이용한 계산

- x_{ij} = 호(i,j) 사이의 흐름량
- $u_{ij}(l_{ij})$ = 호(i,j) 의 용량 상한(하한)값
- f = 최대흐름량
- 목적함수: $Maximize \quad f$
- 제약식: 총 진입흐름량 = 총 진출흐름량
  - $\sum_{\{(i,k)\in A\}} x_{ik} - \sum_{\{(k,j),\in A\}} x_{kj}$ = 순수요량(-f, o, f)
  - $l_{ij} \leq x_{ij} \leq u_{ij}$

최대흐름문제와 최소비용 용량제약 네트워크 문제

최대흐름 문제 예제

최대흐름 문제 예제

❖ 최소비용 용량 제약 네트워크 문제란

- 각 호에 대한 흐름용량제약과 각 마디에서의 공급량 및 수요량을 만족시키면서 총비용을 최소화하는 문제
- 수송문제, 할당문제, 최단경로문제, 최대흐름문제 등 최소비용 용량제약 네트워크 문제의 특수한 경우

❖ 알고리즘을 적용 문제 해결 순서

1. 현실 문제의 네트워크 표현
2. 선형계획법 활용 가능 여부
3. 정수계획법 등 수정 알고리즘 활용 가능 여부
4. 새로운 알고리즘 제작

❖ 선형계획모형을 이용한 계산

- 모형의 종류
- x_{ij} = 마디 **i** 로 부터 마디 **j** 까지의 흐름량
- $u_{ij}(l_{ij})$ = 호(i,j) 의 용량 상한(하한)값
- C_{ij} = 마디 i 로 부터 마디 j 까지의 단위당 흐름 비용
- f_i = 마디 i 에서의 순흐름량
- 목적함수: 각 호의 흐름량과 관련된 총비용
  - $Minimize \quad \sum_{(i,j)} \sum_{A} C_{ij} x_{ij}$
- 제약식:
  - $\sum_{\{(k,j) \in A\}} x_{kj} - \sum_{\{(i,k), \in A\}} x_{ik} = f_k$
  - $l_{ij} \le x_{ij} \le u_{ij}$
- 하한값의 대체
  - $x_{ij} = x'_{ij} + l_{ij}$

$$Minimize \quad \sum_{(i,j)} \sum_{A} C_{ij} x_{ij}$$

$$s.t. \qquad \sum_{\{(k,j) \in A\}} x_{kj} - \sum_{\{(i,k), \in A\}} x_{ik} = f_k$$

$$0 \le x_{ij} \le u_{ij}$$