Logicae 연구회 세미나

매트로이드 이론

Matroid Theory

김민재

경기과학고등학교 43기
Project Logicae 수리정보팀장

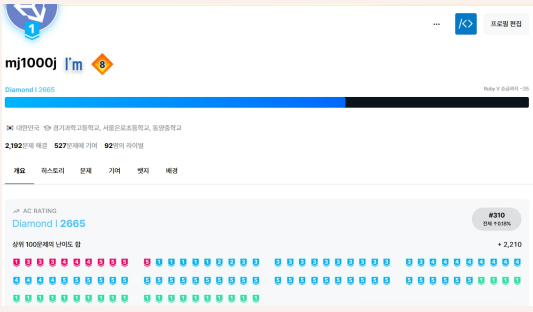
September 27, 2025

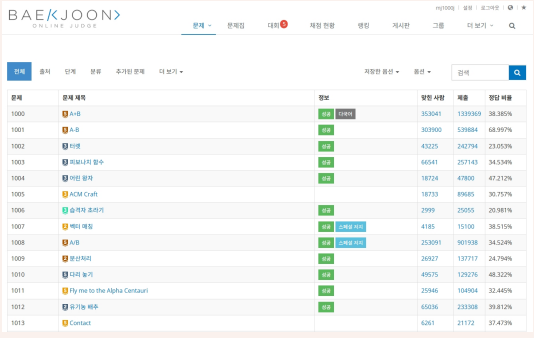
시작하기에 앞서...

여기 앞에 서 있는 사람은 뭐 하는 사람일까요?

시작하기에 앞서...

이런걸 합니다.





시작하기에 앞서...

- solved.ac의 난이도 시스템: 브론즈, 실버, 골드, 플래티넘, 다이아, 루비
- 이 중 가장 어려운 티어인 루비 → '이 알고리즘'만 배우면 루비 10문제를 풀 수 있다!

문제	문제 제목
16046	Rainbow Graph
3836	Coin Collecting
18517	Pick Your Own Nim
18588	Milk Candy
22930	The King's Guards
18890	Seollal
21727	아즈텍의 섬
23052	두 트리
24899	DCMSF
23342	ICPC Kingdom
28012	LaLa and Spirit Summoning
30968	Blume der Liebe
30926	파티

What is matroid?

- Matroid: Matrix에서 파생된 단어로, 벡터공간에서의 일차 독립을 일반화시킨 수학적인 구조

What is matroid?

- Matroid: Matrix에서 파생된 단어로, 벡터공간에서의 일차 독립을 일반화시킨 수학적인 구조
- 그래프 이론, 선형대수학, 조합론, 체론 등의 다양한 분야에서 응용

What is matroid?

- Matroid: Matrix에서 파생된 단어로, 벡터공간에서의 일차 독립을 일반화시킨 수학적인 구조
- 그래프 이론, 선형대수학, 조합론, 체론 등의 다양한 분야에서 응용
- 허준이 교수의 연구 분야 중에서도 매트로이드에 관한 연구 존재

What is matroid?

- Matroid: Matrix에서 파생된 단어로, 벡터공간에서의 일차 독립을 일반화시킨 수학적인 구조
- 그래프 이론, 선형대수학, 조합론, 체론 등의 다양한 분야에서 응용
- 허준이 교수의 연구 분야 중에서도 매트로이드에 관한 연구 존재

- 매트로이드 구조 만족 $\rightarrow$ 그리디 알고리즘 성립

What is matroid?

- Matroid: Matrix에서 파생된 단어로, 벡터공간에서의 일차 독립을 일반화시킨 수학적인 구조
- 그래프 이론, 선형대수학, 조합론, 체론 등의 다양한 분야에서 응용
- 허준이 교수의 연구 분야 중에서도 매트로이드에 관한 연구 존재

- 매트로이드 구조 만족 $\rightarrow$ 그리디 알고리즘 성립
- 매트로이드 교차: 해결이 어려워 보이는 문제를 다항 시간 안에 해결이 가능하도록 하는 강력한 도구

매트로이드의 정의

아래와 같은 조건을 만족하는 구조를 매트로이드로 정의한다.

Definition (매트로이드의 정의)

유한집합 S, 부분집합족 $\mathcal{I} \subseteq 2^S$에 대하여

1. *(I1)* $\emptyset \in \mathcal{I}$
2. *(I2)* $Y \subseteq X \in \mathcal{I} \Rightarrow Y \in \mathcal{I}$
3. *(I3)* $X, Y \in \mathcal{I}, |X| < |Y| \Rightarrow \exists y \in Y \setminus X : X \cup \{y\} \in \mathcal{I}$

위의 세 조건을 만족하는 $\mathcal{M}(S, \mathcal{I})$을 매트로이드라 한다. 이때 S를 Background Set, $\mathcal{I}$의 원소인 집합을 Independent Set이라 한다.

다소 뜬금없어 보이는 위의 정의가 나오게 된 맥락은 무엇일까?

매트로이드의 직관

- S의 모든 부분집합은 Independent 하거나, Dependent 한 것으로 나뉘는데, 그중 Independent 한 것의 모임이 matroid이다.
- 즉, $\mathcal{I}$를 집합족 대신 S의 어떤 부분집합이 Independent한지를 판단하는 "기준"으로 생각할 수 있다.
- 또, 아까의 definition에서 [I1], [I2]만 만족하여도 Independence System이라고 한다.
- 즉, [I3]이 매트로이드의 핵심인 공리라고 볼 수 있다.

(recall) 매트로이드의 정의

- (I3) $X, Y \in \mathcal{I}, |X| < |Y| \Rightarrow \exists y \in Y \setminus X : X \cup \{y\} \in \mathcal{I}$

매트로이드의 예시

대표적인 두 가지의 예시를 통해 매트로이드의 개념을 이해해 보자.

1. Linear Matroid
(선형대수학에서의 '선형 독립')

2. Graphic Matroid
(그래프 이론에서의 'acyclic')

위의 두 가지 예시는 매트로이드에서 사용되는 용어와도 관계가 있는, 가장 직접적인 예시이다.

예시 1: Linear Matroid

- **Background Set (S)**: 어떤 벡터 공간 V의 유한한 벡터의 집합. $S = \{v_1, v_2, \ldots, v_n\}$
- **Independent Sets ($\mathcal{I}$)**: S의 부분집합 중 **선형 독립(linearly independent)**인 것들의 모임.

check

- **(I1)** 공집합 $\emptyset$은 선형 독립이다. (O)
- **(I2)** 선형 독립인 집합의 부분집합은 항상 선형 독립이다. (O)
- **(I3)** X, Y가 선형 독립이고 $|X| < |Y|$이면, Y가 생성하는 공간의 차원이 X가 생성하는 공간의 차원보다 크다. 따라서 Y의 원소 중 X가 생성하는 공간에 포함되지 않는 벡터 y가 존재하며, $X \cup \{y\}$는 선형 독립이다. (O)

선형대수학에서 다루는 '선형 독립'의 개념이 매트로이드에서 그대로 적용되는 것을 알 수 있다.

예시 2: Graphic Matroid

- **Background Set (S)**: 어떤 그래프 $G = (V, E)$의 간선(edge) 집합 E.
- **Independent Sets ($\mathcal{I}$)**: E의 부분집합 중 **사이클을 형성하지 않는(acyclic)** 것들의 모임. (즉, Forest를 이루는 간선 집합)

공리 확인

- **(I1)** 공집합 $\emptyset$은 사이클이 없다. (O)
- **(I2)** 사이클이 없는 간선 집합의 부분집합 또한 사이클이 없다. (O)
- **(I3)** ...?
- 지금까지의 예시와는 달리, Graphic matroid에서 (I3)의 확인은 그다지 자명하지는 않다.

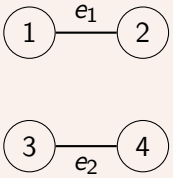
Graphic matroid에서 (I3)의 증명

Proof.

(I3) 증명: X, Y가 사이클 없는 간선 집합(forest)이고 $|X| < |Y|$라고 하자. X는 $k_X = |V| - |X|$개의 컴포넌트를, Y는 $k_Y = |V| - |Y|$개의 컴포넌트를 가진다. $|X| < |Y|$이므로 $k_X > k_Y$이다. 즉, Y는 X보다 컴포넌트 수가 적다. 따라서 Y에는 X에서는 다른 컴포넌트에 속한 두 정점을 연결하는 간선 y가 반드시 존재한다. 이 간선 y를 X에 추가해도 사이클이 생기지 않는다.
$\therefore X \cup \{y\} \in \mathcal{I}$ □

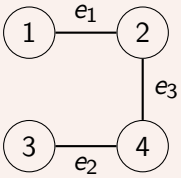
Graphic Matroid 시각화

(I3) 공리를 그림으로 시각화해보자. $|X| = 2, |Y| = 3$



Independent Set X

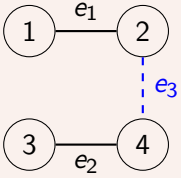
$X = \{e_1, e_2\}$



Independent Set Y

$Y = \{e_1, e_2, e_3\}$

$Y \setminus X = \{e_3\}$이다. X에 e_3를 추가하면...



$X \cup \{e_3\}$은 여전히 사이클이 없다.

Basis

매트로이드의 구조를 더 깊이 이해하기 위해서 몇 가지 용어들을 도입할 것이다.

Definition (Basis의 정의)

Independent set $\mathcal{I}$의 원소 중, 크기가 최대인 집합을 **Basis***라고 한다.*

- *즉, $B \in \mathcal{I}$가 Basis라는 것은, B에 어떤 원소 $e \in S \setminus B$를 추가하더라도 $B \cup \{e\}$는 더 이상 Independent하지 않다는 의미이다. (Maximally Independent)*

눈치가 빠른 사람들이라면, 이 정의에서부터 Basis의 성질도 유추할 수 있을 것이다.

Basis의 성질

Theorem

매트로이드 $\mathcal{M}(S, \mathcal{I})$의 모든 Basis의 크기는 같다.

Proof.

귀류법으로 증명 가능하다. 크기가 다른 두 Basis B_1, B_2가 존재하고, $|B_1| < |B_2|$라고 가정하자. 매트로이드의 (I3) 공리에 의해, B_1에 추가하여 여전히 Independent하게 만들 수 있는 원소 $y \in B_2 \setminus B_1$가 존재한다. 즉, $B_1 \cup \{y\} \in \mathcal{I}$이다. 이는 B_1이 Maximally Independent라는 가정에 모순된다. 따라서 모든 Basis의 크기는 같다. □

Basis의 예시

- **Linear Matroid**:
  - Basis는 벡터 집합 S가 생성하는 벡터 공간(span)의 **기저(basis)**이다.
  - 이는 선형대수학에서 벡터공간의 차원이 정의되는 이유와도 근본적으로 같다.

- **Graphic Matroid**:
  - Basis는 그래프의 **최대 신장 숲(Spanning Forest)**이다.
  - 만약 그래프가 연결되어 있다면(connected), Basis는 **신장 트리(Spanning Tree)**가 된다.
  - 연결 그래프의 모든 신장 트리는 항상 $|V| - 1$개의 간선을 갖고 있음은 자명하고, 이 또한 매트로이드의 성질에 해당한다.

Basis의 다른 정의

사실 매트로이드는 Independent Set이 아닌 Basis나 Circuit을 통해 정의할 수도 있다. Basis를 이용한 정의는 다음과 같다.

Definition (Basis Axioms)

유한집합 S, 공집합이 아닌 부분집합족 $\mathcal{B} \subseteq 2^S$가 아래 두 조건을 만족하면, $\mathcal{B}$는 어떤 매트로이드의 Basis 집합이 된다.

1. *(B1)* $\mathcal{B} \neq \emptyset$
2. *(B2)* $B_1, B_2 \in \mathcal{B}$*이고* $x \in B_1 \setminus B_2$*이면,* $y \in B_2 \setminus B_1$*가 존재하여* $(B_1 \setminus \{x\}) \cup \{y\} \in \mathcal{B}$*를 만족한다. (Basis Exchange Property)*

(B2) 교환 공리는 한 Basis에서 원소 하나를 빼면, 다른 Basis의 적절한 원소 하나를 가져와 새로운 Basis를 만들 수 있다는 의미이다. 이는 선형대수학의 기저(Basis) 교환 정리와 동일하다.

Circuit의 도입

Basis를 이해했으니, 이제 Circuit을 알아보자.

질문

어떤 Basis B가 있고, 여기에 속하지 않는 원소 $e \in S \setminus B$가 있다고 하자. $B \cup \{e\}$는 Maximally Independent였던 B의 성질에 의해 무조건 Dependent Set이 될 것이다.

그렇다면, $B \cup \{e\}$는 어떤 구조를 가질까?

Theorem (Fundamental Circuit)

Basis B와 $e \in S \setminus B$에 대해, $B \cup \{e\}$는 단 하나의 유일한 **Circuit***을 포함한다. 이를 e의 B에 대한 fundamental circuit이라 부르고 $C(e, B)$로 표기한다.*

Circuit의 도입

Basis를 이해했으니, 이제 Circuit을 알아보자.

질문

어떤 Basis B가 있고, 여기에 속하지 않는 원소 $e \in S \setminus B$가 있다고 하자. $B \cup \{e\}$는 Maximally Independent였던 B의 성질에 의해 무조건 Dependent Set이 될 것이다.

그렇다면, $B \cup \{e\}$는 어떤 구조를 가질까?

Theorem (Fundamental Circuit)

Basis B와 $e \in S \setminus B$에 대해, $B \cup \{e\}$는 단 하나의 유일한 **Circuit***을 포함한다. 이를 e의 B에 대한 fundamental circuit이라 부르고 $C(e, B)$로 표기한다.*

- Graphic Matroid 예시: 스패닝 트리(Basis)에 간선 하나를 추가하면 정확히 하나의 사이클(Circuit)이 생기는 것과 같다.

주요 개념: Circuit

Definition (Circuit의 정의)

Dependent set 중, 크기가 최소인 집합을 **Circuit***이라고 한다.*

- *즉, $C \subseteq S$가 Circuit이라는 것은, C는 Dependent하지만, C에서 어떤 원소를 하나라도 제거하면 그 즉시 Independent해진다는 의미이다. (Minimally Dependent)*

예시

- **Linear Matroid**: Circuit은 **최소 선형 종속** 집합이다. 벡터들이 선형 종속이지만, 하나라도 빼면 선형 독립이 되는 상태이다.
- **Graphic Matroid**: Circuit은 그래프의 **단순 사이클(Simple Cycle)** 그 자체이다.

Circuit Axioms

역시, Circuit으로도 매트로이드를 정의할 수 있다.

Definition (Circuit Axioms)

유한집합 S, 부분집합족 $\mathcal{C} \subseteq 2^S$가 아래 세 조건을 만족하면, $\mathcal{C}$는 어떤 매트로이드의 Circuit 집합이 된다.

1. *(C1)* $\emptyset \notin \mathcal{C}$
2. *(C2)* $C_1, C_2 \in \mathcal{C}$*이고* $C_1 \subseteq C_2$*이면,* $C_1 = C_2$*이다. (서로를 포함하지 않음)*
3. *(C3)* $C_1, C_2 \in \mathcal{C}$*이고* $C_1 \neq C_2$*,* $e \in C_1 \cap C_2$*이면,* $C_3 \subseteq (C_1 \cup C_2) \setminus \{e\}$*를 만족하는* $C_3 \in \mathcal{C}$*가 존재한다. (Circuit Elimination)*

(C3)이 조금 복잡해 보일 수 있다. 두 개의 다른 사이클이 한 점(원소)에서 만날 때, 그 점을 제외한 나머지 부분에 또 다른 사이클이 있다는 의미로 해석할 수 있다.

주요 개념: Rank

마지막 핵심 개념은 Rank(계수) 함수이다.

Definition (Rank의 정의)

집합 $A \subseteq S$에 대하여, A의 부분집합 중 가장 큰 Independent set의 크기를 A의 **Rank***라 하고, $r(A)$로 표기한다.*

$$r(A) = \max\{|X| : X \subseteq A, X \in \mathcal{I}\}$$

- Rank 함수는 주어진 집합이 '얼마나 독립적인지'를 측정하는 척도라고 볼 수 있다.
- 매트로이드 전체의 랭크 $r(S)$는 Basis의 크기와 같다.
- $r(A) = |A|$ 라면, A는 그 자체로 Independent Set이라는 뜻이다.

Rank 함수의 성질

Rank 함수는 다음과 같은 중요한 성질들을 만족한다.

1. $0 \leq r(A) \leq |A|$
2. (Monotonicity) $A \subseteq B \implies r(A) \leq r(B)$
3. (Submodularity) $r(A \cup B) + r(A \cap B) \leq r(A) + r(B)$

Submodularity(준모듈성)

세 번째 성질인 Submodularity가 Rank 함수의 핵심이다. 두 집합의 랭크 합이, 각 집합의 랭크 합보다 작거나 같다는 의미이다. 이는 'Diminishing returns'(수확 체감)의 개념과 비슷하다. 즉, 이미 독립적인 원소들을 많이 가진 집합에 새로운 원소를 추가해도 랭크가 잘 증가하지 않는다는 것이다.

이 Rank 함수로도 매트로이드를 정의할 수 있다. 결국 Independent set, Basis, Circuit, Rank는 모두 매트로이드라는 큰 그림의 다른 표현일 뿐이다.

Greedy Algotihm for Matroid

- 지금까지 매트로이드의 정의와 개념에 대해서 다뤄보았다.
- 그렇다면, 실제로 매트로이드가 알고리즘 문제 해결 분야(PS)에서 어떻게 활용될 수 있을까?

다음 문제를 생각해 보자.

최대 가중치 기저 문제

만약 S의 각 원소 e에 가중치(weight) $w(e)$가 부여되어 있다면, 어떻게 해야 가중치의 합이 최대(또는 최소)가 되는 Basis를 찾을 수 있을까?

예시: 그래프의 간선에 가중치가 있을 때, 가중치 합이 최소가 되는 스패닝 트리를 찾는 문제.

Greedy Algotihm for Matroid

가장 쉽게 생각할 수 있는 방법은 탐욕 알고리즘(greedy algorithm)이다.

1. 매 순간, 지금 당장 가장 좋아 보이는 선택을 한다.
2. 이전의 선택은 번복하지 않으며, 앞으로의 선택에 대해서도 고려하지 않는다.

가중치 합이 최대인 Basis를 찾는 Greedy 전략

1. S의 모든 원소를 가중치가 큰 순서대로 정렬한다.
2. 처음에는 빈 집합 $A = \emptyset$에서 시작한다.
3. 정렬된 순서대로 원소 e를 하나씩 확인하면서, $A \cup \{e\}$가 여전히 Independent하다면, e를 A에 추가한다.
4. 모든 원소를 다 확인할 때까지 반복한다.

이 단순한 방법이 항상 최적의 해를 찾아줄까? 일반적으로는 그렇지 않지만...

매트로이드와 탐욕 알고리즘

놀랍게도, 문제 상황이 매트로이드 구조를 만족한다면 앞선 그리디 전략은 항상 최적해를 찾을 수 있음이 보장된다.

Theorem (Rado-Edmonds Theorem)

어떤 Independence System $(S, \mathcal{I})$에 대해, $(S, \mathcal{I})$가 매트로이드라면 위에서 설명한 탐욕 알고리즘이 임의의 가중치 함수에 대해 항상 최적의 해(가중치 합이 최대인 Basis)를 찾는다.

탐욕적 전략의 유효성

왜 매트로이드에서는 그리디한 전략이 유효할까?

- Greedy로 찾은 해를 $A = \{a_1, \ldots, a_r\}$, 최적 해를 $B = \{b_1, \ldots, b_r\}$라고 하자. ($w(a_1) \geq \cdots \geq w(a_r)$, $w(b_1) \geq \cdots \geq w(b_r)$)

탐욕적 전략의 유효성

왜 매트로이드에서는 그리디한 전략이 유효할까?

- Greedy로 찾은 해를 $A = \{a_1, \ldots, a_r\}$, 최적 해를 $B = \{b_1, \ldots, b_r\}$라고 하자. ($w(a_1) \geq \cdots \geq w(a_r)$, $w(b_1) \geq \cdots \geq w(b_r)$)
- 만약 $A \neq B$라면, A와 B가 달라지는 첫 번째 원소가 존재한다. 즉, 어떤 k에 대해 $a_i = b_i$ ($i < k$), $a_k \neq b_k$.

탐욕적 전략의 유효성

왜 매트로이드에서는 그리디한 전략이 유효할까?

- Greedy로 찾은 해를 $A = \{a_1, \ldots, a_r\}$, 최적 해를 $B = \{b_1, \ldots, b_r\}$라고 하자. ($w(a_1) \geq \cdots \geq w(a_r)$, $w(b_1) \geq \cdots \geq w(b_r)$)
- 만약 $A \neq B$라면, A와 B가 달라지는 첫 번째 원소가 존재한다. 즉, 어떤 k에 대해 $a_i = b_i$ ($i < k$), $a_k \neq b_k$.
- Greedy의 선택 방식에 의해, 알고리즘은 b_k보다 a_k를 먼저 고려했거나, b_k와 가중치가 같았을 것이므로 $w(a_k) \geq w(b_k)$이다.

탐욕적 전략의 유효성

왜 매트로이드에서는 그리디한 전략이 유효할까?

- Greedy로 찾은 해를 $A = \{a_1, \ldots, a_r\}$, 최적 해를 $B = \{b_1, \ldots, b_r\}$라고 하자. $(w(a_1) \geq \cdots \geq w(a_r),\ w(b_1) \geq \cdots \geq w(b_r))$
- 만약 $A \neq B$라면, A와 B가 달라지는 첫 번째 원소가 존재한다. 즉, 어떤 k에 대해 $a_i = b_i$ $(i < k)$, $a_k \neq b_k$.
- Greedy의 선택 방식에 의해, 알고리즘은 b_k보다 a_k를 먼저 고려했거나, b_k와 가중치가 같았을 것이므로 $w(a_k) \geq w(b_k)$이다.
- **(I3) 교환 공리** 이용: $X = \{a_1, \ldots, a_{k-1}\}$는 Independent. $Y = \{b_1, \ldots, b_k\}$도 Independent. $|X| < |Y|$ 이므로, X에 추가해도 Independent를 유지하는 $b_j \in Y \setminus X$가 존재한다.

탐욕적 전략의 유효성

왜 매트로이드에서는 그리디한 전략이 유효할까?

- Greedy로 찾은 해를 $A = \{a_1, \ldots, a_r\}$, 최적 해를 $B = \{b_1, \ldots, b_r\}$라고 하자. $(w(a_1) \geq \cdots \geq w(a_r),\ w(b_1) \geq \cdots \geq w(b_r))$
- 만약 $A \neq B$라면, A와 B가 달라지는 첫 번째 원소가 존재한다. 즉, 어떤 k에 대해 $a_i = b_i$ $(i < k)$, $a_k \neq b_k$.
- Greedy의 선택 방식에 의해, 알고리즘은 b_k보다 a_k를 먼저 고려했거나, b_k와 가중치가 같았을 것이므로 $w(a_k) \geq w(b_k)$이다.
- **(I3) 교환 공리** 이용: $X = \{a_1, \ldots, a_{k-1}\}$는 Independent. $Y = \{b_1, \ldots, b_k\}$도 Independent. $|X| < |Y|$ 이므로, X에 추가해도 Independent를 유지하는 $b_j \in Y \setminus X$가 존재한다.
- Exchange Argument $\rightarrow$ 귀납적으로 증명

최소 신장 트리(MST) 문제

앞선 논의를 바탕으로, 그래프의 최소 신장 트리를 구하는 알고리즘을 제시할 수 있다.

- **문제**: 그래프 $G = (V, E)$와 간선별 가중치 $w(e)$가 주어졌을 때, 간선 가중치의 합이 최소인 Spanning Tree를 찾아라.
- 이 문제는 **Graphic Matroid** 위에서의 최적화 문제이다.
- 가중치 최소화는 -(가중치)의 최대화와 같으므로, 아래의 알고리즘을 적용할 수 있다.

MST를 위한 Greedy 알고리즘 (Kruskal's Algorithm)

1. 모든 간선을 가중치가 **작은** 순서대로 정렬한다.
2. 빈 집합 $A = \emptyset$에서 시작한다.
3. 정렬된 간선 e를 순서대로 확인하며, $A \cup \{e\}$가 사이클을 만들지 않으면 A에 추가한다.
4. 간선을 $|V| - 1$개 선택할 때까지 반복한다.

우리는 방금 MST를 구하는 크루스칼 알고리즘을 재창조했다!

결론 및 요약

- **매트로이드**는 선형 독립, 그래프의 acyclic 등 '독립'의 개념을 추상화한 수학적 구조이다.
- Independent Set, Basis, Circuit, Rank 등 다양한 방식으로 정의될 수 있지만, 본질은 같다.
- 이러한 논의를 가능하게 하는 핵심은 **(I3) 교환 공리**로, 작은 independent set은 항상 더 큰 independent set의 원소를 가져와 크기를 키울 수 있다.
- 요약하면, 매트로이드는 **탐욕 알고리즘(Greedy Algorithm)의 정당성을 보장**하는 구조라는 것이다.
- 어떤 문제가 매트로이드 구조임을 보일 수만 있다면, 그리디한 접근법으로 최적해를 찾을 수 있다.

매트로이드 교차 (Matroid Intersection)

새로운 문제

만약 우리가 **두 가지 종류의 독립 조건**을 동시에 만족하는 가장 큰 집합을 찾고 싶다면 어떻게 해야 할까?

Definition (매트로이드 교차 문제)

동일한 Background Set S를 갖는 두 매트로이드 $\mathcal{M}_1 = (S, \mathcal{I}_1)$와 $\mathcal{M}_2 = (S, \mathcal{I}_2)$가 주어졌을 때,

$$I \in \mathcal{I}_1 \cap \mathcal{I}_2 = \{A \subseteq S \mid A \in \mathcal{I}_1 \text{ and } A \in \mathcal{I}_2\}$$

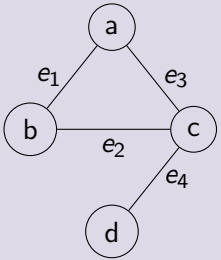
를 만족하면서 크기 $|I|$가 최대인 집합 I는 무엇인가?

이때 $\mathcal{I}_1 \cap \mathcal{I}_2$의 원소를 **공통 독립 집합(Common Independent Set)**이라 하며, 위의 문제는 공통 독립 집합의 원소 중 크기가 최대인 원소를 찾는 문제라 할 수 있다.

매트로이드 교차 문제의 어려움

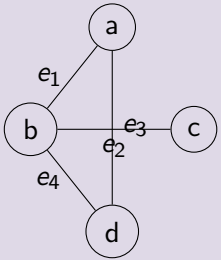
- 혹시 $\mathcal{M}_{12} = (S, \mathcal{I}_1 \cap \mathcal{I}_2)$가 매트로이드는 아닐까?
- 안타깝게도, 두 매트로이드의 교차는 일반적으로 매트로이드가 아니다.

반례: Graphic Matroid의 교차



그래프 G1

$\mathcal{M}_1 = M(G_1)$

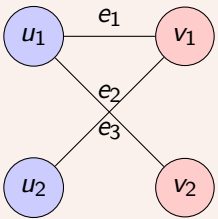


그래프 G2

$\mathcal{M}_2 = M(G_2)$

예시 1: 이분 매칭 (Bipartite Matching)

매트로이드 교차의 가장 대표적인 예시이이다.



- **문제**: 이분 그래프에서 크기가 가장 큰 매칭(matching)을 찾아라.
- 어떤 간선 집합이 매칭이 되려면, 모든 정점의 차수(degree)가 1 이하여야 한다.
  - 조건 1: 왼쪽 정점들의 차수가 1 이하.
  - 조건 2: 오른쪽 정점들의 차수가 1 이하.

  이 두 조건을 각각 매트로이드로 모델링할 수 있다.

매트로이드를 통한 이분 매칭의 표현

- S: 그래프의 간선(edge) 집합 E.
- **매트로이드** $\mathcal{M}_1$: 왼쪽 정점 U에 대한 제약
  - $\mathcal{I}_1 = \{A \subseteq E \mid \forall u \in U, \deg_A(u) \leq 1\}$
  - 어떤 간선 집합이 왼쪽 정점을 두 번 이상 사용하지만 않으면 independent
- **매트로이드** $\mathcal{M}_2$: 오른쪽 정점 V에 대한 제약
  - $\mathcal{I}_2 = \{A \subseteq E \mid \forall v \in V, \deg_A(v) \leq 1\}$
  - 어떤 간선 집합이 오른쪽 정점을 두 번 이상 사용하지만 않으면 independent

즉, 최대 이분 매칭 문제는 $\mathcal{M}_1$과 $\mathcal{M}_2$의 매트로이드 교차 문제로 이해할 수 있다.

알고리즘의 아이디어: 증가 경로

- 매트로이드 교차 문제의 대표적인 예시로 최대 이분 매칭 문제를 알아보았다.
- 혹시, 일반적인 매트로이드 교차 문제에서 이분 매칭 알고리즘처럼 **증가 경로 (Augmenting Path)**를 찾는 아이디어를 사용해 볼 수 있지 않을까?

이분 매칭으로부터 추론한 매트로이드 교차 문제 해결 알고리즘

1. 빈 공통 독립 집합 $I = \emptyset$에서 시작한다.
2. 현재 I의 크기를 1만큼 늘릴 수 있는 '증가 경로' P를 찾는다.
3. 만약 경로 P를 찾았다면, I를 $I \Delta P$로 갱신하고 2번으로 돌아간다.
4. 더 이상 증가 경로가 없다면, 현재 I가 최대 크기의 공통 독립 집합이다.

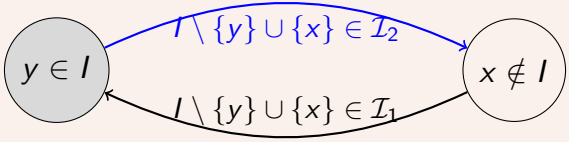
아직 위의 알고리즘의 정당성을 증명하지는 않았지만, 일단 믿음을 가져보자.
그런데, 매트로이드 교차 문제에서의 '증가 경로'는 무엇일까?

교환 그래프 (Exchange Graph)

증가 경로를 찾기 위해, 현재 공통 독립 집합 I에 대한 **교환 그래프** $D(I)$를 정의한다.

Definition (교환 그래프 $D(I)$)

- 정점**(Vertex)**: *Background set S의 모든 원소.*
- 간선**(Arc)**: *$x, y \in S$에 대하여,*
  1. *$x \in S \setminus I, y \in I$ 이고 $(I \setminus \{y\}) \cup \{x\} \in \mathcal{I}_1$ 이면, (x, y) 간선을 추가한다.*
     ($\mathcal{M}_1$에서 y를 빼고 x를 넣어도 독립이 유지되는가?)
  2. *$x \in I, y \in S \setminus I$ 이고 $(I \setminus \{x\}) \cup \{y\} \in \mathcal{I}_2$ 이면, (x, y) 간선을 추가한다.*
     ($\mathcal{M}_2$에서 x를 빼고 y를 넣어도 독립이 유지되는가?)



증가 경로의 정의

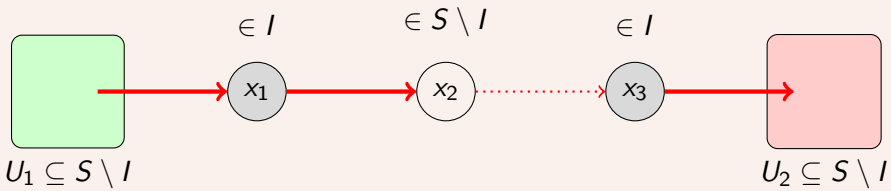
교환 그래프를 만들었으니, 이제 증가 경로를 정의할 수 있다.

Definition (증가 경로 (Augmenting Path))

현재 공통 독립 집합 I에 대하여,

- $U_1 = \{x \in S \setminus I \mid I \cup \{x\} \in \mathcal{I}_1\}$
- $U_2 = \{y \in S \setminus I \mid I \cup \{y\} \in \mathcal{I}_2\}$

라 하자. (U_1, U_2는 I에 그냥 추가해도 각각 $\mathcal{M}_1, \mathcal{M}_2$에서 독립성을 유지하는 원소들)
교환 그래프 $D(I)$에서 U_1에 속한 정점에서 시작하여 U_2에 속한 정점에서 끝나는 최단 경로를 증가 경로라고 한다.



매트로이드 교차의 핵심 보조정리

Theorem

공통 독립 집합 I는 최대 크기 $\iff$ 교환 그래프 $D(I)$에 증가 경로가 존재하지 않는다.

이 정리에서 ($\Rightarrow$) 방향, 즉 "I가 최대 크기이면, 증가 경로가 없다"의 대우 명제를 증명해보자.

매트로이드 교차의 핵심 보조정리

"교환 그래프 $D(I)$에 **증가 경로 P가 존재한다면**, I는 **최대 크기가 아니다**."

결국, 증가 경로 P를 찾으면, 현재 I보다 더 큰 공통 독립 집합 I'을 만들 수 있다는 것을 보이면 충분하다.

매트로이드 교차의 핵심 보조정리

대칭차 (Symmetric Difference)

증가 경로 P가 존재할 때, 새로운 집합 I'을 다음과 같이 구성하자.

$$I' = I \Delta V(P) = (I \setminus V(P)) \cup (V(P) \setminus I)$$

크기가 1만큼 증가하는 이유

- 증가 경로 $P = (x_0, x_1, \ldots, x_k)$는 $x_0 \in U_1$, $x_k \in U_2$에서 시작하고 끝난다.
- 즉, 경로의 시작과 끝 원소 x_0, x_k는 I에 속하지 않는다.
- 경로는 I 밖의 원소와 안의 원소를 번갈아 지난다: $x_{2i} \notin I, x_{2i+1} \in I$.
- 그렇기 때문에 경로에 있는 원소들 중 I에 속하지 않는 것이 I에 속하는 것보다 항상 1개 더 많다.
- 즉, $|I'| = |I| - |I \cap V(P)| + |V(P) \setminus I| = |I| + 1$ 이 성립한다.

증명 개요: $I' \in \mathcal{I}_1$

$P = (x_0, x_1, \ldots, x_k)$를 증가 경로라 하자.

- $x_0 \in U_1$, $x_k \in U_2$
- $x_{2i} \in S \setminus I$, $x_{2i+1} \in I$

$I' = (I \setminus \{x_1, x_3, \ldots\}) \cup \{x_0, x_2, \ldots\}$

1. 먼저, $I \cup \{x_0\} \in \mathcal{I}_1$ 이다. ($x_0 \in U_1$의 정의)
2. 교환 그래프의 간선 (x_2, x_1)의 정의에 의해, $(I \setminus \{x_1\}) \cup \{x_2\} \in \mathcal{I}_1$ 이다.
3. 매트로이드 (I3) 공리를 여러 번 적용하면, $I \cup \{x_0\}$에서 x_1을 빼고 x_2를 더한 집합, 즉 $(I \setminus \{x_1\}) \cup \{x_0, x_2\}$도 $\mathcal{I}_1$에 속함을 보일 수 있다.
4. 이 과정을 경로의 끝까지 반복하면, 결국 $I' \in \mathcal{I}_1$임을 증명할 수 있다.

(상세 증명은 복잡하여 생략하지만, (I3) 공리를 연쇄적으로 적용하는 방식으로 이해해할 수 있다.)

증명 개요: $I' \in \mathcal{I}_2$

이번에는 경로의 끝에서부터 시작하자. $P = (x_0, x_1, \ldots, x_k)$

1. 먼저, $I \cup \{x_k\} \in \mathcal{I}_2$ 이다. ($x_k \in U_2$의 정의)
2. 교환 그래프의 간선 (x_{k-1}, x_k)의 정의에 의해, $(I \setminus \{x_{k-1}\}) \cup \{x_k\} \in \mathcal{I}_2$ 이다.
3. (I3) 공리를 비슷하게 연쇄적으로 적용하면, $I \cup \{x_k\}$에서 x_{k-1}을 빼고 x_{k-2}를 더하는 식으로 거꾸로 진행할 수 있다.
4. 최종적으로 $I' = (I \setminus \{x_1, x_3, \ldots\}) \cup \{x_0, x_2, \ldots\}$가 $\mathcal{I}_2$에 속함을 증명할 수 있다.

결론

증가 경로 P가 존재하면, $I' = I \Delta V(P)$는 크기가 1 더 큰 새로운 공통 독립 집합이다.

매트로이드 교차 알고리즘

이제 전체 알고리즘을 아래와 같이 정리할 수 있다.

Edmonds' Matroid Intersection Algorithm

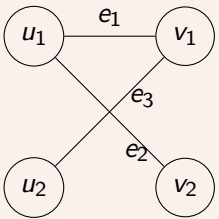
1. $I \leftarrow \emptyset$.
2. **Loop**:
   1. 현재 I에 대한 교환 그래프 $D(I)$를 구성한다.
   2. 집합 $U_1 = \{x \notin I \mid I \cup \{x\} \in \mathcal{I}_1\}$과 $U_2 = \{y \notin I \mid I \cup \{y\} \in \mathcal{I}_2\}$를 찾는다.
   3. BFS(너비 우선 탐색)를 사용하여 $D(I)$에서 U_1의 정점에서 U_2의 정점으로 가는 최단 경로 P를 찾는다.
   4. 만약 경로 P가 존재하지 않으면, 현재 I를 최적해로 반환하고 종료한다.
   5. 경로 P가 존재하면, $I \leftarrow I \Delta V(P)$로 갱신하고 루프의 처음으로 돌아간다.

이 알고리즘은 다항 시간 안에 최대 공통 독립 집합을 찾아준다.

예제: 이분 매칭 다시 보기 (1/5)

알고리즘을 직접 따라가 보자.

그래프



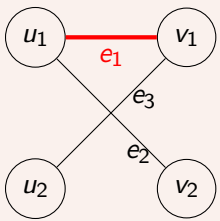
초기 상태

- $I_0 = \emptyset$
- $S \setminus I_0 = \{e_1, e_2, e_3\}$
- $\mathcal{M}_1$(U-side): $I_0 \cup \{e\}$는 모든 $e \in S \setminus I_0$에 대해 독립. $\implies U_1 = \{e_1, e_2, e_3\}$.
- $\mathcal{M}_2$(V-side): $I_0 \cup \{e\}$는 모든 $e \in S \setminus I_0$에 대해 독립. $\implies U_2 = \{e_1, e_2, e_3\}$.

증가 경로: U_1에서 U_2로 가는 경로는 자기 자신으로 가는 경로. e.g., $P_0 = (e_1)$.

업데이트: $I_1 = I_0 \Delta \{e_1\} = \{e_1\}$.

예제: 이분 매칭 다시 보기 (2/5)



현재 매칭 $I_1 = \{e_1\}$

상태

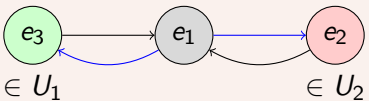
- $S \setminus I_1 = \{e_2, e_3\}$
- U_1: $I_1 \cup \{e_2\}$는 u_1에서 충돌. $I_1 \cup \{e_3\}$는 독립. $\implies$ $U_1 = \{e_3\}$.
- U_2: $I_1 \cup \{e_2\}$는 독립. $I_1 \cup \{e_3\}$는 v_1에서 충돌. $\implies$ $U_2 = \{e_2\}$.

교환 그래프 $D(I_1)$에서 $U_1 = \{e_3\}$ 에서 $U_2 = \{e_2\}$ 로 가는 경로를 찾아야 한다.

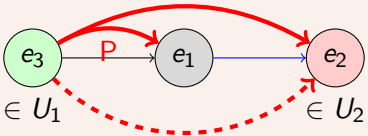
예제: 이분 매칭 다시 보기 (3/5)

교환 그래프 $D(I_1)$ 구성

- 정점: $\{e_1, e_2, e_3\}$
- 간선 (Type 1: $x \notin I, y \in I$ for $\mathcal{M}_1$)
  - $x = e_2, y = e_1$: $I_1 \setminus \{e_1\} \cup \{e_2\} = \{e_2\}$. $\mathcal{M}_1$에서 독립 $\implies$ (e_2, e_1) 간선 존재.
  - $x = e_3, y = e_1$: $I_1 \setminus \{e_1\} \cup \{e_3\} = \{e_3\}$. $\mathcal{M}_1$에서 독립 $\implies$ (e_3, e_1) 간선 존재.
- 간선 (Type 2: $x \in I, y \notin I$ for $\mathcal{M}_2$)
  - $x = e_1, y = e_2$: $I_1 \setminus \{e_1\} \cup \{e_2\} = \{e_2\}$. $\mathcal{M}_2$에서 독립 $\implies$ (e_1, e_2) 간선 존재.
  - $x = e_1, y = e_3$: $I_1 \setminus \{e_1\} \cup \{e_3\} = \{e_3\}$. $\mathcal{M}_2$에서 독립 $\implies$ (e_1, e_3) 간선 존재.



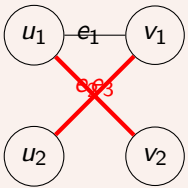
예제: 이분 매칭 다시 보기 (4/5)



증가 경로: $U_1 = \{e_3\}$ 에서 $U_2 = \{e_2\}$ 로 가는 최단 경로 $P_1 = (e_3, e_1, e_2)$.

업데이트: $I_2 = I_1 \Delta V(P_1) = \{e_1\} \Delta \{e_3, e_1, e_2\} = (\{e_1\} \setminus \{e_1\}) \cup (\{e_2, e_3\} \setminus \emptyset) = \{e_2, e_3\}$.

새로운 매칭 $I_2 = \{e_2, e_3\}$



예제: 이분 매칭 다시 보기 (5/5)

현재 매칭 $I_2 = \{e_2, e_3\}$

- $S \setminus I_2 = \{e_1\}$
- U_1: $I_2 \cup \{e_1\}$은 u_1에서 충돌. $\implies U_1 = \emptyset$.
- U_2: $I_2 \cup \{e_1\}$은 v_1에서 충돌. $\implies U_2 = \emptyset$.

U_1이 공집합이므로, 더 이상 증가 경로는 존재하지 않는다.

결론

알고리즘이 종료되고, 최대 매칭은 $I_2 = \{e_2, e_3\}$이며 크기는 2이다.

매트로이드 교차의 쌍대성 정리

Theorem (Edmonds' Matroid Intersection Theorem)

두 매트로이드 $\mathcal{M}_1 = (S, \mathcal{I}_1)$, $\mathcal{M}_2 = (S, \mathcal{I}_2)$에 대하여, 최대 공통 독립 집합의 크기는 다음과 같다.

$$\max_{I \in \mathcal{I}_1 \cap \mathcal{I}_2} |I| = \min_{U \subseteq S} (r_1(U) + r_2(S \setminus U))$$

여기서 r_1, r_2는 각각 $\mathcal{M}_1, \mathcal{M}_2$의 랭크 함수이다.

- 이 정리는 네트워크 flow의 Max-flow Min-cut 정리, 이분 매칭의 Konig의 정리와 같은 맥락에 있다.
- 이는 매트로이드 교차 알고리즘의 정당성을 증명해 준다.
- 위의 정리에 대한 구체적인 증명은, MIT 18.455 강의의 lecture note를 참고하라.

결론: 매트로이드 교차

- **매트로이드 교차**는 두 개의 독립성 조건을 동시에 만족하는 가장 큰 집합을 찾는 문제이다.
- 이분 매칭, 유향 그래프의 Arborescence 등 수많은 문제가 매트로이드 교차로 모델링될 수 있다.
- 적절한 모델링을 통해 일반적으로는 다항 시간 안에 해결하기 어려울 것 같은 문제를 다항 시간에 해결할 수 있다.

참고 자료 (References)

- **Papers & Articles**
  - Jack Edmonds. "Submodular functions, matroids, and certain polyhedra." *Combinatorial structures and their applications*, 1970.
  - Eugene L. Lawler. "Matroid intersection algorithms." *Mathematical Programming*, 1975.

- **Online Resources (Blogs & Lecture Notes)**
  - lighton's blog: Matroid (1) (`https://lighton.tistory.com/12`)
  - lighton's blog: Matroid Intersection (2) (`https://lighton.tistory.com/14`)
  - Codeforces Blog: Yet another blog about matroids (`https://codeforces.com/blog/entry/58202`)
  - SSM InfoStack: Introduction to Matroid (`https://infossm.github.io/blog/2019/05/15/introduction-to-matroid/`)
  - SSM InfoStack: Matroid Intersection (`https://infossm.github.io/blog/2019/06/17/Matroid-Intersection/`)
  - MIT 18.455 (Goemans): Advanced Combinatorial Opt. (`https://math.mit.edu/ goemans/18455-2020SP.html`)

감사합니다

Q & A