

#3. 그리디 + 맵트로이드

나정휘 (jhnah917)

<https://justicehui.github.io/>

그리디 예시 - 11

예시 11 (예시 9의 가중치 없는 버전)

- 수행하는데 1초가 걸리는 작업이 N개 있음
- i번째 작업은 D[i]초 안에 끝내야 함
- 수행할 수 있는 작업 수의 최댓값을 구하는 문제
- 단순한 전략
 - 정답 집합 R을 공집합으로 초기화
 - 주어진 작업을 아무 순서대로 보면서, 만약 R + {i번째 작업}을 모두 수행할 수 있다면
 - 집합 R에 i번째 작업 추가
 - 정말 이 전략이 항상 올바른 답을 구할까?



```
bool feasible(set, task){  
    // set + {task}에 있는  
    // 모든 작업을 수행할 수 있는지 판별  
}  
for(int i=0; i<N; i++){  
    if(feasible(R, i)) R.insert(i);  
}
```

그리디 예시 - 11

예시 11

- “더 이상 원소를 추가할 수 없는” 집합들의 크기가 모두 같음을 보이면 됨
- 귀류법을 사용해서, $|A| < |B|$ 인 두 집합 A, B 모두 더 이상 추가할 수 있는 원소가 없다고 가정하자.
 - $|A| < |B|$ 이므로 B 에서는 작업을 하지만 A 에서는 작업을 하지 않는 시간 $[t, t + 1)$ 이 존재함
 - $B - A$ 의 원소 중 $[t, t + 1)$ 구간에 끼워넣을 수 있는 작업이 항상 존재함
 - 따라서, A 에 추가할 수 있는 원소가 없다는 가정에 모순이 생김 $\rightarrow$ maximal한 집합의 크기는 모두 같음
- maximal한 집합의 크기가 모두 같음
 - 원소를 추가할 수 있을 때 추가하는 것을 반복하면 항상 크기가 최대인 집합을 구할 수 있음
- 한 가지 추가 수확
 - 만약 두 집합 A, B 모두 유효한 집합이면서 $|A| < |B|$ 이면, $A \cup \{x\}$ 도 유효한 집합이 되는 $x \in B \setminus A$ 가 존재함

그리디 예시 - 9

예시 9

- 수행하는데 1초가 걸리는 작업이 N 개 있음
- i 번째 작업은 $D[i]$ 초 안에 끝내야 하며, 마감 기한 안에 작업을 완료하면 $V[i]$ 원을 받을 수 있음
- $V[i] > 0$ 일 때, 얻을 수 있는 금액을 최대화하는 문제
- 간단한 전략
 - 정답 집합 R 을 공집합으로 초기화
 - 주어진 작업을 가중치($V[i]$) 내림차순으로 보면서, 만약 $R + \{i\text{번째 작업}\}$ 을 모두 수행할 수 있다면
 - 집합 R 에 i 번째 작업 추가
 - 정말 이 전략이 항상 가중치의 합을 최대화하는 집합을 구할까?

그리디 예시 - 9

예시 9

- 일단 $V[i] > 0$ 이므로 정답은 maximal 집합 중 하나이고, 그리디 전략은 maximal 집합을 잘 찾음
- 귀류법을 사용해서, 그리디 전략이 구한 집합 A 보다 더 좋은 집합 B 가 존재한다고 가정하자.
 - $V[i] > 0$ 이므로 B 도 A 와 크기가 같은 maximal 집합이라고 가정해도 된다.
 - A, B 의 원소들을 각각 가중치 내림차순으로 $A = \{a_1, a_2, \dots, a_s\}, B = \{b_1, b_2, \dots, b_s\}$ 이라고 하자
 - $w(A) < w(B)$ 이므로 처음으로 $w(a_k) < w(b_k)$ 가 되는 지점 $1 \leq k \leq s$ 가 존재
 - $A_{k-1} = \{a_1, a_2, \dots, a_{k-1}\}, B_k = \{b_1, b_2, \dots, b_{k-1}, b_k\}$ 라고 하자
 - 작업을 몇 개 제거한다고 유효하지 않은 집합이 되진 않으므로, A, B 의 부분 집합인 A_{k-1}, B_k 도 유효한 집합임
 - $|A_{k-1}| < |B_k|$ 이므로 $A_{k-1} \cup \{b_i\}$ 도 유효한 집합이 되는 $b_i \in B_k \setminus A_{k-1}$ 이 존재함 ($1 \leq i \leq k$)
 - $w(a_k) < w(b_k) \leq w(b_i)$ 이므로 그러한 b_i 가 존재했다면 그리디 알고리즘은 a_k 대신 b_i 를 선택했을 것임
 - 따라서 A 보다 더 좋은 해 B 가 존재한다면, A 는 그리디 알고리즘의 결과물이 될 수 없음
- 한 가지 추가 수확
 - 만약 집합 A 가 유효한 집합이라면, $A' \subset A$ 도 유효한 부분 집합임

그리디 예시 - 12

예시 12 (최대 신장 트리)

- 연결 무향 가중치 그래프 $G = (V, E)$ 가 주어지면, 모든 정점이 연결되도록 간선을 몇 개 선택하는 문제
- 이때 선택한 간선은 트리를 이루어야 하며, 가중치의 합을 최대화해야 함
- 크루스칼 알고리즘
 - 간선을 가중치 내림차순으로 정렬한 뒤
 - 만약 간선을 추가하더라도 사이클이 없는 그래프(포레스트)임이 유지된다면
 - 간선을 정답 집합에 추가
- 알고리즘의 정당성을 다시 증명해 보자.

그리디 예시 - 12

예시 12 (최대 신장 트리)

- 먼저, 모든 maximal한 집합의 크기가 같음을 보이자.
- 귀류법을 사용해서, $|E_A| < |E_B|$ 인 두 집합 E_A, E_B 모두 더 이상 추가할 수 있는 원소가 없다고 가정하자.
 - 두 포레스트 $G_A = (V, E_A), G_B = (V, E_B)$ 를 생각해 보자
 - G_A 는 $|V| - |E_A|$ 개, G_B 는 $|V| - |E_B|$ 개의 트리로 구성되어 있으므로 G_B 가 더 적은 개수의 트리를 갖고 있음
 - 따라서 G_B 의 어떤 트리 T 는 G_A 에서 서로 다른 트리에 속한 두 정점 u, v 를 동시에 포함하고 있음
 - 따라서 T 의 $u - v$ 경로에 속한 간선 중 최소한 하나는 $|E_A|$ 에 속하지 않음
 - 그 간선을 E_A 에 추가하더라도 사이클이 생기지 않음 $\rightarrow E_A$ 에 추가할 수 있는 원소가 없다는 가정에 모순 발생
- maximal한 집합의 크기가 모두 같음
 - 원소를 추가할 수 있을 때 추가하는 것을 반복하면 항상 크기가 최대인 집합을 구할 수 있음
 - 우리는 그러한 집합이 "신장 트리"라는 것을 이미 알고 있음
- 한 가지 추가 수확
 - 만약 두 간선 집합 E_A, E_B 모두 포레스트면서 $|E_A| < |E_B|$ 이면
 - $E_A \cup \{e\}$ 도 포레스트가 되는 간선 $e \in E_B \setminus E_A$ 가 존재함

그리디 예시 - 12

예시 12 (최대 신장 트리)

- 이제, 크루스칼 알고리즘이 항상 가중치 합이 최대인 신장 트리를 구함을 증명하자.
- 귀류법을 사용해서, 크루스칼 알고리즘이 구한 집합 A 보다 더 좋은 집합 B 가 존재한다고 가정하자.
 - A, B 모두 신장 트리이므로 두 집합은 크기가 같은 간선 집합이다.
 - A, B 의 원소들을 각각 가중치 내림차순으로 $A = \{a_1, a_2, \dots, a_s\}, B = \{b_1, b_2, \dots, b_s\}$ 이라고 하자
 - $w(A) < w(B)$ 이므로 처음으로 $w(a_k) < w(b_k)$ 가 되는 지점 $1 \leq k \leq s$ 가 존재
 - $A_{k-1} = \{a_1, a_2, \dots, a_{k-1}\}, B_k = \{b_1, b_2, \dots, b_{k-1}, b_k\}$ 라고 하자
 - 간선을 몇 개 제거한다고 사이클이 생기지 않으므로, A, B 의 부분 집합인 A_{k-1}, B_k 도 포레스트임
 - $|A_{k-1}| < |B_k|$ 이므로 $A_{k-1} \cup \{b_i\}$ 도 포레스트가 되는 $b_i \in B_k \setminus A_{k-1}$ 이 존재함 ($1 \leq i \leq k$)
 - $w(a_k) < w(b_k) \leq w(b_i)$ 이므로 그러한 b_i 가 존재했다면 크루스칼 알고리즘은 a_k 대신 b_i 를 선택했을 것임
 - 따라서 A 보다 더 좋은 해 B 가 존재한다면, A 는 크루스칼 알고리즘의 결과물이 될 수 없음
- 한 가지 추가 수확
 - 만약 집합 A 가 포레스트라면, $A' \subset A$ 도 포레스트임

매트로이드

매트로이드 (Matroid)

- 앞에서 본 두 문제의 풀이 증명 과정은 단어 몇 개를 제외하면 완전히 똑같다!
- 이런 구조를 일반화할 수 있을까?
- 우리가 사용한 성질을 다시 정리해 보면...
 - 어떤 유한 집합 S 이 있고, 특정 조건을 만족하는 S 의 부분 집합을 구하는 문제
 - 조건을 만족하는 집합들의 집합을 $I \subseteq 2^S$ 라고 하면,
 - $B \in I$ 이고 $A \subseteq B$ 이면 $A \in I$.
 - $A, B \in I$ 이고 $|A| < |B|$ 이면, $A \cup \{x\} \in I$ 를 만족하는 $x \in B \setminus A$ 가 존재함.
- 위 조건을 만족하는 구조에서는
 - 크기가 최대인 집합을 찾을 수 있고
 - 가중치 합이 최대인 집합을 찾을 수 있고
 - (가중치 오름차순으로 정렬하면) 가중치 합이 최소인 집합도 찾을 수 있다.

매트로이드

매트로이드 (Matroid)

- 아래 네 가지 조건을 만족하는 순서쌍 $M = (S, I)$ 를 "매트로이드", I 의 원소들을 "독립 집합"이라고 부름
 - S 는 유한 집합
 - $I \subseteq 2^S$, 즉, I 는 S 의 부분 집합들로 구성된 집합
 - $\emptyset \in I$. 즉, 공집합은 독립 집합이어야 함
 - (상속성) $B \in I$ 이고 $A \subseteq B$ 이면 $A \in I$
 - (확장성) $A \in I, B \in I, |A| < |B|$ 이면 $A \cup \{x\} \in I$ 를 만족하는 $x \in B \setminus A$ 가 존재함
- 확장(extension) / 기저(basis)
 - $A \in I$ 이고 $x \notin A$ 일 때 $A \cup \{x\} \in I$ 이면 x 를 A 의 "확장"이라고 부름
 - 만약 $A \in I$ 의 확장이 존재하지 않으면 A 를 "기저"라고 부름
 - 즉, $A \in I$ 이고 $A \subsetneq B \in I$ 인 B 가 없으면 A 는 기저

매트로이드

매트로이드 (Matroid)

- 예시 1 – Graphic Matroid
 - $M_G = (S, I)$ 에서 S 를 무향 그래프의 간선 집합, I 를 사이클이 존재하지 않는 간선 집합으로 정의하자
 - 즉, I 는 forest를 이루는 간선 집합
 - 예시 12에서 보았듯이, M_G 는 매트로이드
 - G 가 연결 그래프면 M_G 의 기저는 스패닝 트리
- 예시 2 – Uniform Matroid
 - 어떤 집합 S 가 있을 때, 크기가 k 이하인 모든 부분 집합들을 I 로 정의하면 $M = (S, I)$ 는 매트로이드
- 예시 3 – Partition Matroid
 - S 의 분할 $S_1, S_2, \dots, S_m$ 과 양수 $k_1, k_2, \dots, k_m$ 이 있을 때, $I = \{X \subseteq S; \forall i |X \cap S_i| \leq k_i\}$ 로 정의하면 매트로이드
 - S_i 의 원소는 최대 k_i 개만 선택해야 함을 의미함
- 예시 4 – Cographic Matroid
 - 무향 그래프 $G = (V, E)$ 에서 $S = E, I = \{X \subseteq E, E - X \text{가 연결 그래프}\}$ 로 정의하면 매트로이드

매트로이드

매트로이드 (Matroid)

- Theorem 1. 매트로이드에 있는 모든 기저들의 크기는 동일함
 - A 가 매트로이드 $M = (S, I)$ 의 기저이면서 $|A| < |B|$ 인 기저 B 가 존재한다고 가정하자
 - 확장성에 의해 $A \cup \{x\} \in I, x \in B \setminus A$ 를 만족하는 x 가 존재할 텐데, 이는 A 가 기저라는 것에 모순
- 따라서 원소를 추가할 수 있을 때마다 계속 추가하면 항상 최대 크기의 집합을 얻을 수 있음
 - Local Optimum = Global Optimum 이므로 그리디하게 원소를 선택할 수 있음

매트로이드

매트로이드 (Matroid)

- 가중치 매트로이드
 - 매트로이드 $M = (S, I)$ 에서 S 의 각 원소 x 가 가중치 $w(x)$ 를 갖는 상황
 - 집합 X 의 가중치는 $w(X) = \sum_{x \in X} w(x)$, 즉 원소들의 가중치 합으로 정의
 - 예시 12(최소 신장 트리): graphic matroid에서 가중치가 최소인 기저를 찾는 문제
 - 예시 9(마감 시간과 보상이 있는 단위 시간 작업 스케줄링): 가중치가 최대인 독립 집합을 찾는 문제
- 최대 가중치 독립 집합을 찾는 방법
 - 가중치가 큰 원소부터 차례대로 보면서, 독립 집합에 추가할 수 있으면 추가하는 전략
- Theorem 2. 이 그리디 알고리즘은 음수 가중치가 없을 때 항상 최대 가중치 독립 집합을 찾음
 - 최소 가중치 독립 집합은 가중치가 작은 원소부터 보면 됨

매트로이드

매트로이드 (Matroid)

- Theorem 2. 이 그리디 알고리즘은 음수 가중치가 없을 때 항상 최대 가중치 독립 집합을 찾음
 - 그리디 알고리즘으로 구한 독립 집합 A 보다 더 좋은 독립 집합 B 가 존재한다고 가정하자
 - Theorem 1에 의해 A 는 기저이고, 모든 가중치가 0 이상이므로 B 도 기저라고 생각해도 됨
 - A, B 의 원소들을 각각 가중치 내림차순으로 $A = \{a_1, a_2, \dots, a_n\}, B = \{b_1, b_2, \dots, b_n\}$ 이라고 하자
 - $w(A) < w(B)$ 이므로 처음으로 $w(a_k) < w(b_k)$ 가 되는 지점 $1 \leq k \leq n$ 이 존재
 - $A_{k-1} = \{a_1, a_2, \dots, a_{k-1}\}, B_k = \{b_1, b_2, \dots, b_{k-1}, b_k\}$ 라고 하자
 - $A_{k-1} \in I, B_k \in I, |A_{k-1}| < |B_k|$ 이므로 $A_{k-1} \cup \{b_i\} \in I$ 를 만족하는 $b_i \in B_k \setminus A_{k-1}$ 이 존재함 ($1 \leq i \leq k$)
 - $w(a_k) < w(b_k) \leq w(b_i)$ 이므로 A 보다 더 좋은 해 B 가 있었다면 그리디 알고리즘은 a_k 대신 b_i 를 선택함
 - 따라서 A 는 그리디 알고리즘의 결과물이 될 수 없음

매트로이드

매트로이드 (Matroid)

- 만약 문제에서 주어진 집합과 조건이 매트로이드 구조를 만족한다면
 - 어떤 집합 X 가 독립 집합인지 확인하는 연산을 구현한 뒤
 - 이 연산을 $|S|$ 번 호출해서 최대 크기 / 최대 가중치 독립 집합을 찾을 수 있다!
- 이러한 연산을 oracle이라고 부름

매트로이드

더 알아보면 좋은 것

- Matroid Intersection

- 같은 ground set에서 정의된 두 매트로이드 $M_1 = (S, I_1), M_2 = (S, I_2)$ 가 있다고 하자.
- 두 매트로이드에서 모두 독립 집합인 가장 큰 (or 가중치가 가장 큰) 독립 집합을 찾는 문제
- S 의 크기에 대한 다항 횟수 번의 oracle 호출로 해결할 수 있음

- 이분 그래프 매칭

- 이분 그래프 $G = (L \cup R, E)$ 에서 간선들이 서로 끝점을 공유하지 않도록 간선을 선택하는 문제
- L (또는 R)에서 각 정점은 최대 한 번 선택되어야 함 $\rightarrow$ partition matroid
- 두 매트로이드에서 모두 독립 집합인 집합을 구하는 문제

- 유향 신장 트리 (Arborescence)

- 유향 그래프 $G = (V, E)$ 에서 아래 조건을 만족하는 부분 그래프 T 를 찾는 문제
 - 간선의 방향을 무시했을 때, T 에 사이클이 없어야 함
 - 모든 정점의 in degree는 최대 1이어야 함
- 사이클 조건 $\rightarrow$ graphic matroid
- in degree 조건 $\rightarrow$ partition matroid

매트로이드

더 알아보면 좋은 것

- Matroid Intersection

- 같은 ground set에서 정의된 두 매트로이드 $M_1 = (S, I_1), M_2 = (S, I_2)$ 가 있다고 하자.
- 두 매트로이드에서 모두 독립 집합인 가장 큰 (or 가중치가 가장 큰) 독립 집합을 찾는 문제
- S 의 크기에 대한 다항 횟수 번의 oracle 호출로 해결할 수 있음

- 하지만 세 매트로이드의 intersection은 $P \neq NP$ 라면 다항 시간에 구할 수 없음

- 해밀턴 경로

- 방향 그래프가 주어지면, 모든 정점을 정확히 한 번씩 방문하는 길이가 $N-1$ 인 경로를 찾는 문제
- NP-Complete 문제임. 즉, 이 문제가 다항 시간에 풀리면 $P = NP$ 임.

- 해밀턴 경로에서 모든 정점의 in degree는 1 이하 $\rightarrow$ partition matroid
- 해밀턴 경로에서 모든 정점의 out degree는 1 이하 $\rightarrow$ partition matroid
- 사이클이 없어야 함 $\rightarrow$ graphic matroid

- 만약 세 매트로이드의 최대 공통 독립 집합을 다항 시간에 구할 수 있다면
- 그 크기가 $N-1$ 인지 확인하는 것으로 해밀턴 회로 문제를 다항 시간에 해결할 수 있음