

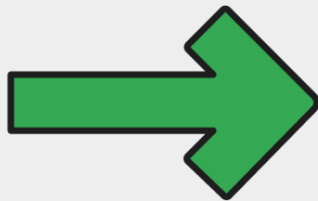


Google Developer Group  
Editable University Name

# POMO: Policy Optimization with Multiple Optima for Reinforcement Learning

GDGoC INU AI Part Paper Seminar

AI Core 박희선

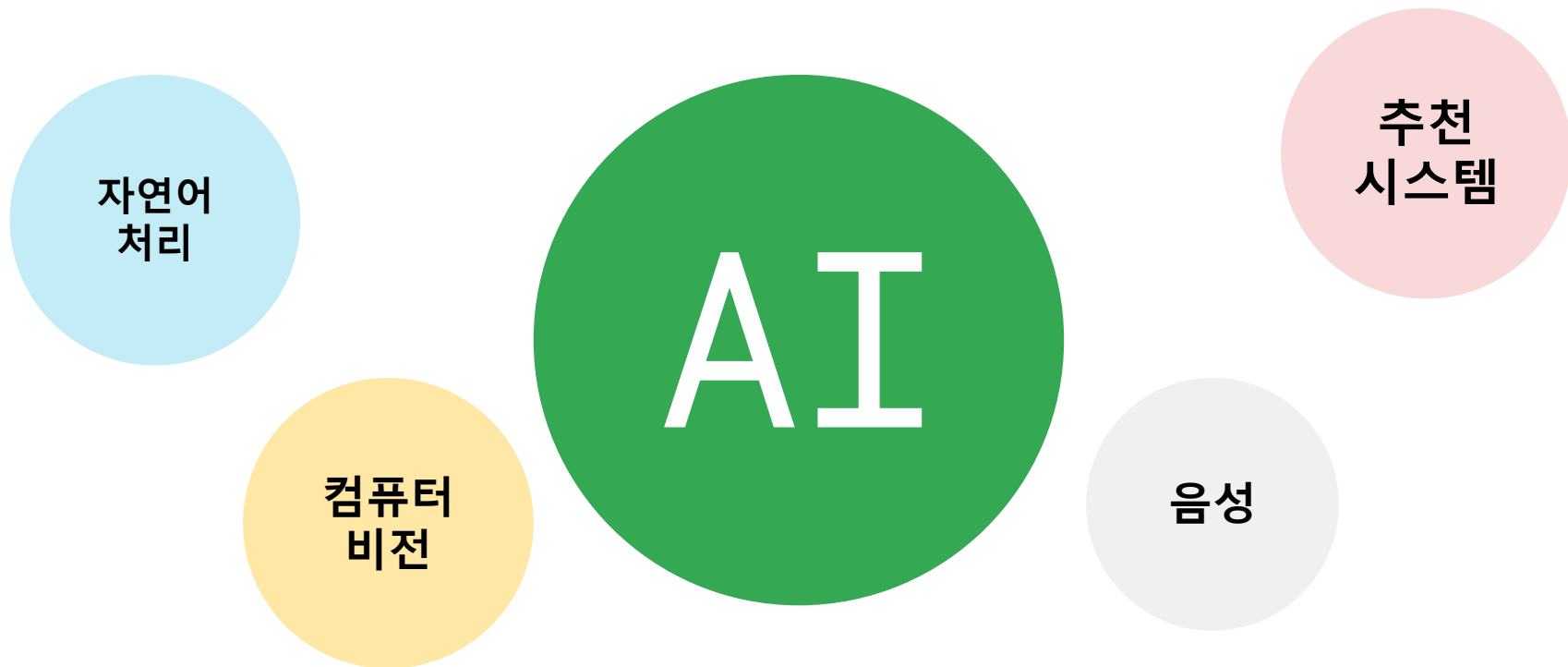


# Contents

1. Background
2. Introduction
3. Method
4. Experiments
5. Conclusion



# 1. Background



# 1. Background

# Combinatorial Optimization

조합최적화

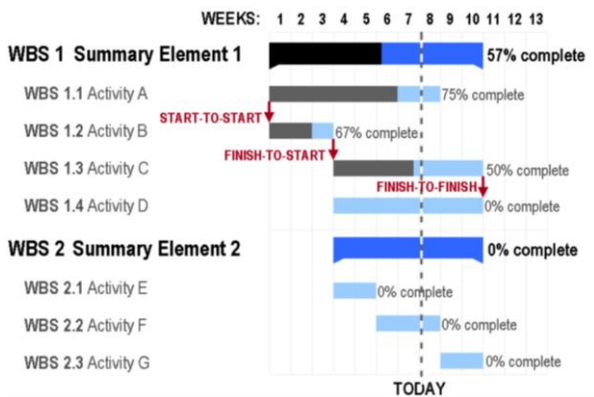


# 1. Background

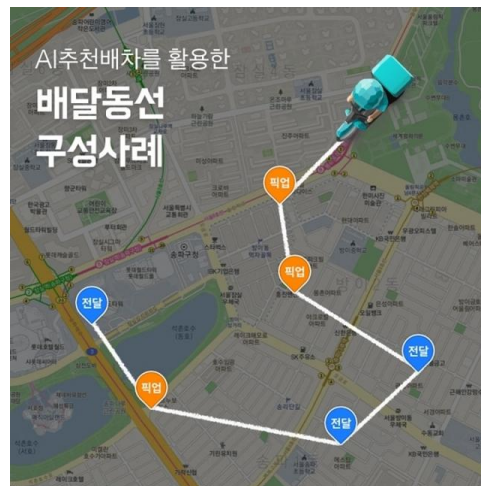
## 조합 최적화란?

- 유한한 후보해 집합에서 목적함수를 최적화하는 조합을 찾는 것

### 일정 관리 문제



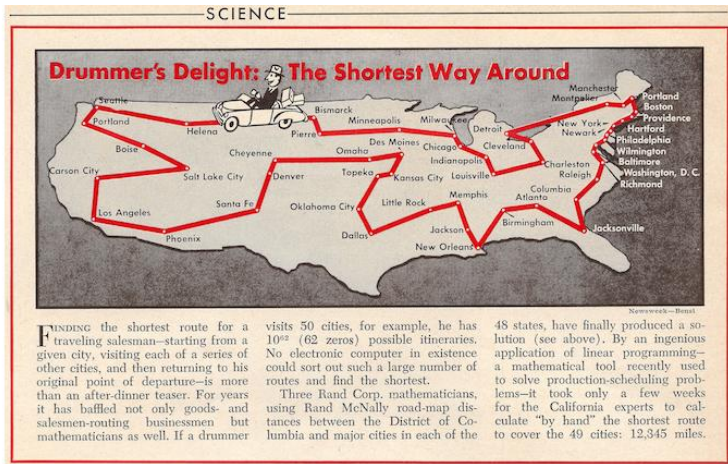
### 최소 경로 문제



# 1. Background

조합최적화 문제 예시?

TSP (Traveling Salesman Problem): 판매원이  $n$ 개의 도시를 한 번 씩 방문하여 최단거리로 순회하는 문제



$n$ 개의 도시

→  $n!$  만큼의 가능한 조합이 생성됨

10개의 도시

→ 3,628,800 만큼의 가능한 조합이 생성됨

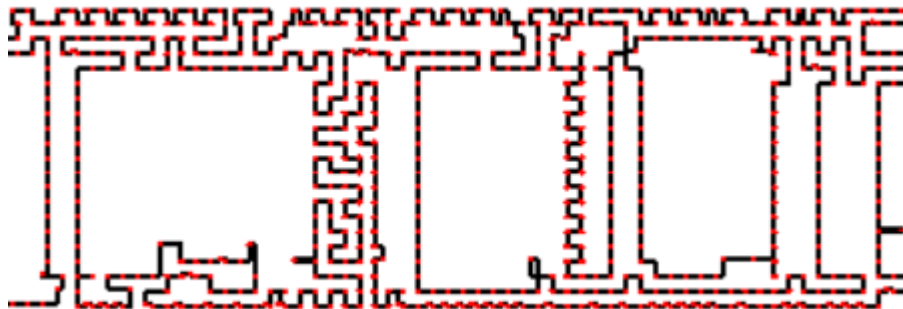
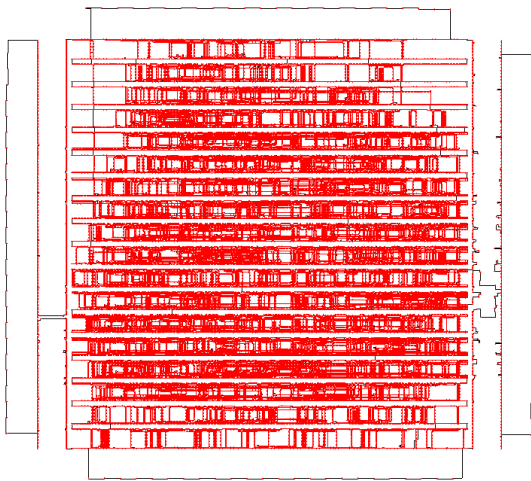


# 1. Background

VSLI (Very Large Scale Integrated): 초대규모 집적회로

실제로, 총 85,900개의 노드로 구성된 TSP문제를 Concorde 알고리즘으로 풀

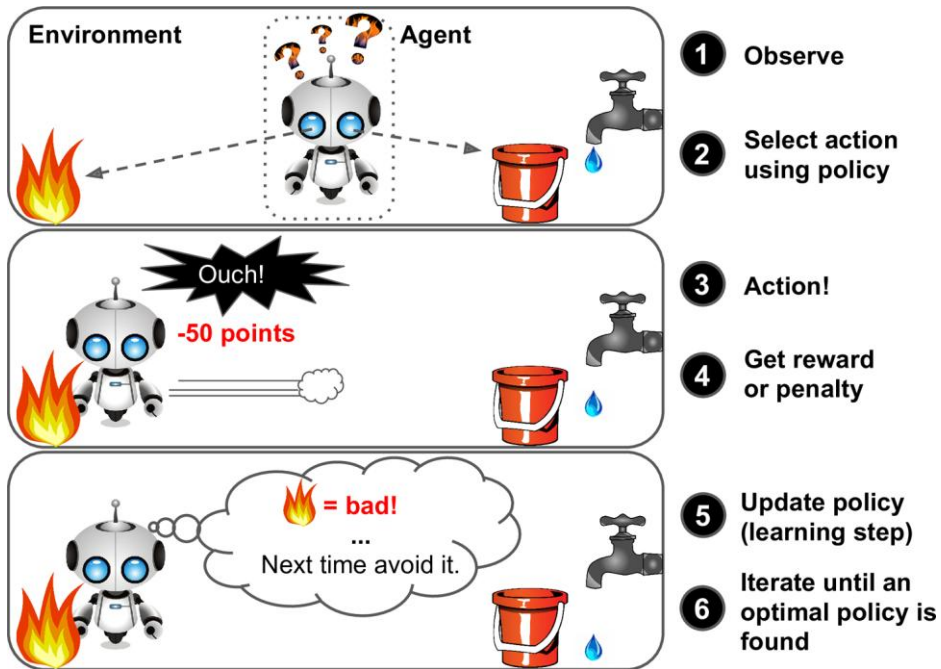
→ 해당 문제에 맞는 알고리즘을 따로 마련해야하고 문제를 푸는 시간도 긴 단점이 존재



# 1. Background

## 강화학습이란?

- 주어진 상황에서 어떠한 행동을 취할지 시행착오를 통해 학습하는 것





## 2. Introduction

알고리즘을 이용한 풀이는 문제는 규모확장성, 일반화, 병렬 연산 등의 면에서 불리  
→ 강화학습을 이용한 조합최적화 문제를 푸는 것을 목표로 함



CO 문제의 대칭성을 활용해 여러 개의 롤아웃을 생성하여 병렬 실행함



low-variance baseline을 통해 지역최적화 문제를 피함



greedy rollout 기법을 사용하여 효과적으로 추론



## 2. Introduction

본 논문에서 실험한 조합최적화 문제

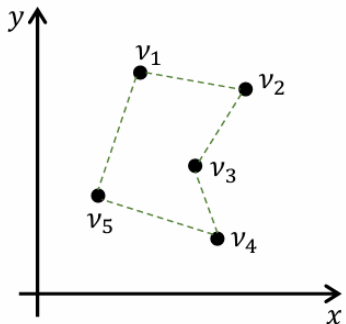
1. TSP (Traveling Salesman Problem, 외판원 문제)
2. CVRP (Capacitated Vehicle Routing Problem, 차량 경로 문제)
3. KP (Knapsack Problem, 배낭문제)



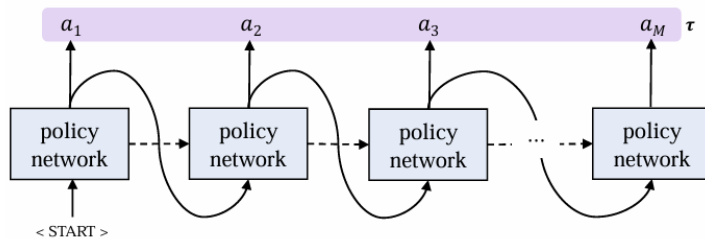
# 3. Method

## 🔍 Explorations from multiple starting nodes

- 조합최적화 문제에선 ( $v_1, v_2, v_3, v_4, v_5$ )로 이루어진 고리와 ( $v_2, v_3, v_4, v_5, v_1$ )으로 이루어진 경로는 차이가 없음
- 즉, 시작점에 크게 영향을 안 받음
- 그러나 기존의 방식에서는 첫 번째 시작 노드도 Token으로서 학습에 포함이 됨



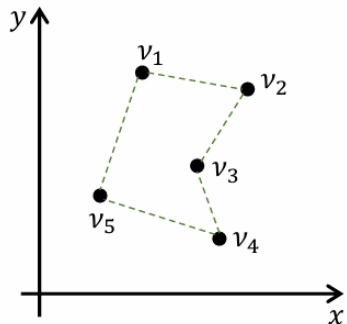
<기존의 작동 방식>



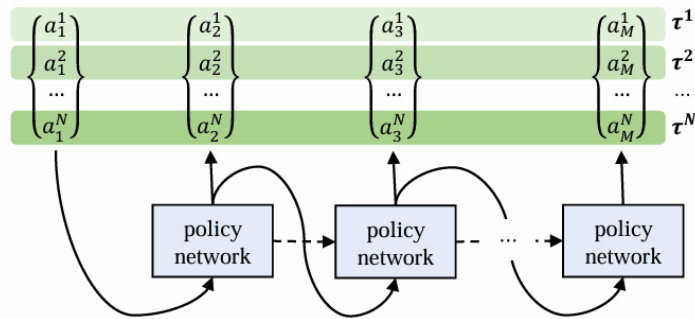
# 3. Method

## 🔍 Explorations from multiple starting nodes

- 노드의 개수만큼 서로 다른 시작 노드를 만듦
- n개의 해답 경로를 생성하여 다양한 관점에서의 경로를 만들도록 강제함
- 즉, 기존에는 한 문제에 대해서 완벽해질 때까지 같은 방식으로 반복해 푸는 거라면 POMO는 한 문제에 대해서 여러 방식으로 반복해 푸는 것



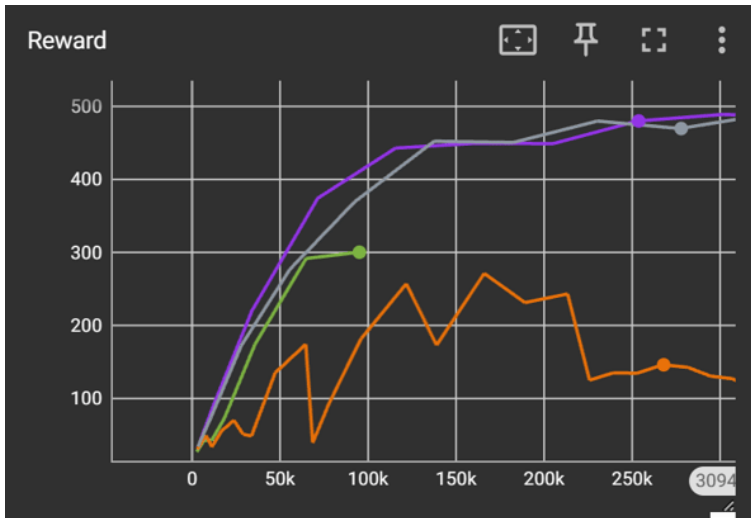
<POMO의 작동 방식>



# 3. Method

## 🔍 A shared baseline for policy gradients

- 강화학습에서 baseline의 중요성



Run	Value	Step	Time	Relative
runs\reinforce_logs\REINFORCE_BASELINE	480	2,53,724	8/4/24, 2:22 PM	7.325 min
runs\reinforce_logs\REINFORCE_BASELINE_2	469.7	2,77,879	8/5/24, 7:24 PM	9.609 min
runs\reinforce_logs\REINFORCE_NO_BASELINE	300.2	95,070	8/4/24, 2:16 PM	1.849 min
runs\reinforce_logs\REINFORCE_NO_BASELINE_2	146.1	2,67,930	8/5/24, 6:22 PM	8.127 min




# 3. Method

## 🔍 A shared baseline for policy gradients

- 생성된 여러 개의 경로에 대해 동일한 baseline을 사용하기 때문에 보상의 변동성이 줄어들
- Baseline의 분산이 낮아져 학습의 안정성이 향상
- 여러 경로의 평균을 baseline으로 산정하기 때문에 지역최적해로의 조기 수렴을 방지

baseline의 분산이 낮아지면 업데이트의 안정성이 높아짐!


$$\nabla_{\theta} J(\theta) \approx \frac{1}{N} \sum_{i=1}^N (R(\boldsymbol{\tau}^i) - b^i(s)) \nabla_{\theta} \log p_{\theta}(\boldsymbol{\tau}^i | s)$$

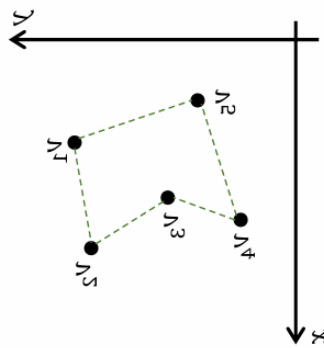
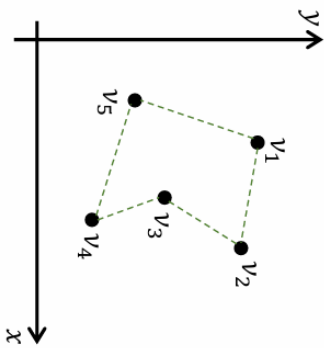
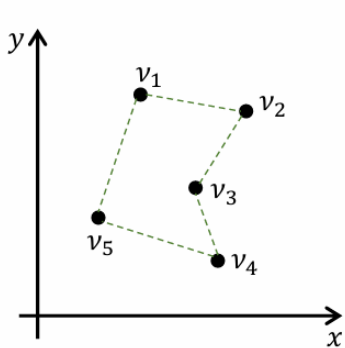
$$b^i(s) = b_{\text{shared}}(s) = \frac{1}{N} \sum_{j=1}^N R(\boldsymbol{\tau}^j) \quad \text{for all } i.$$



# 3. Method

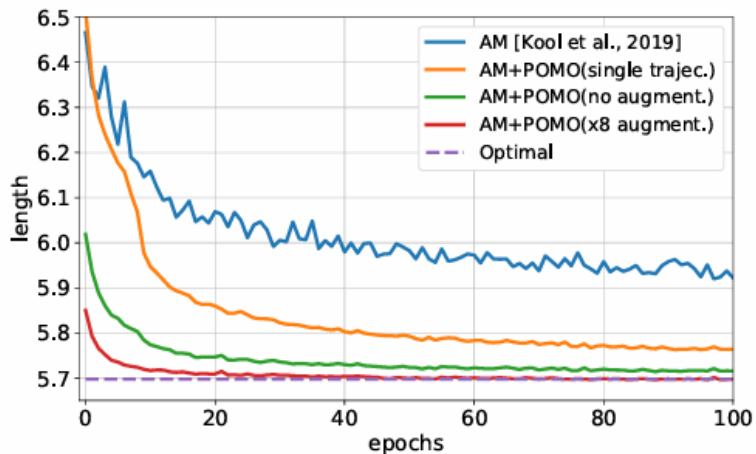
## 🔍 Multiple greedy trajectories for inference

- 추론 시 일반적으로 샘플링 기법을 사용하여 여러 번 반복해 최적의 해를 찾음
- POMO는 시작 지점을 여러 개로 두어 경로를 만들고, 해당 경로는 탐욕적으로(경정적으로) 최적의 해를 찾음
- 단 POMO는 노드의 개수 만큼 시작지점의 개수가 제한되는 단점이 있어 Instance Augmentation을 통해 추가적인 탐색이 가능하도록 함

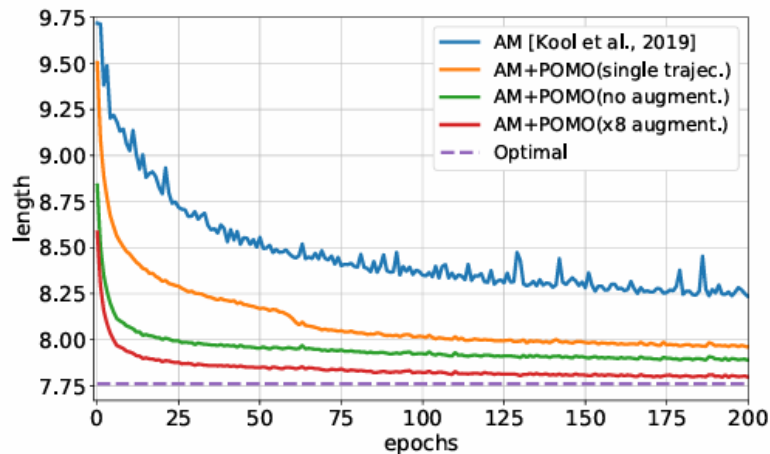


## 4. Experiments

기존의 방식(파란 선)보다 POMO(빨간, 초록, 노란 색)가 최단거리에 더 빠르게 도달



(a) TSP50



(b) TSP100





## 4. Experiments

- Optimal과의 차이가 비교적 적음
- 최적해에 다가갈 때까지의 시간도 짧음

Table 2: Experiment results on TSP

Method	TSP20			TSP50			TSP100		
	Len.	Gap	Time	Len.	Gap	Time	Len.	Gap	Time
Concorde	3.83	-	(5m)	5.69	-	(13m)	7.76	-	(1h)
LKH3	3.83	0.00%	(42s)	5.69	0.00%	(6m)	7.76	0.00%	(25m)
Gurobi	3.83	0.00%	(7s)	5.69	0.00%	(2m)	7.76	0.00%	(17m)
OR Tools	3.86	0.94%	(1m)	5.85	2.87%	(5m)	8.06	3.86%	(23m)
Farthest Insertion	3.92	2.36%	(1s)	6.00	5.53%	(2s)	8.35	7.59%	(7s)
GCN [9], beam search	3.83	0.01%	(12m)	5.69	0.01%	(18m)	7.87	1.39%	(40m)
Improv. [11], {5000}	3.83	0.00%	(1h)	5.70	0.20%	(1h)	7.87	1.42%	(2h)
Improv. [12], {2000}	3.83	0.00%	(15m)	5.70	0.12%	(29m)	7.83	0.87%	(41m)
AM [10], greedy	3.84	0.19%	( $\ll$ 1s)	5.76	1.21%	(1s)	8.03	3.51%	(2s)
AM [10], sampling	3.83	0.07%	(1m)	5.71	0.39%	(5m)	7.92	1.98%	(22m)
POMO, single trajec.	3.83	0.12%	( $\ll$ 1s)	5.73	0.64%	(1s)	7.84	1.07%	(2s)
POMO, no augment.	3.83	0.04%	( $\ll$ 1s)	5.70	0.21%	(2s)	7.80	0.46%	(11s)
POMO, $\times 8$ augment.	3.83	0.00%	(3s)	5.69	0.03%	(16s)	7.77	0.14%	(1m)



## 5. Conclusion



심층 강화학습을 기반으로 휴리스틱을 사용하지 않았음에도 최적에 가까운 성능을 보임



추론 속도를 단축시킴



다양한 종류의 조합최적화 문제에 대해 일반화된 방법



# Thank you!



Google Developer Group