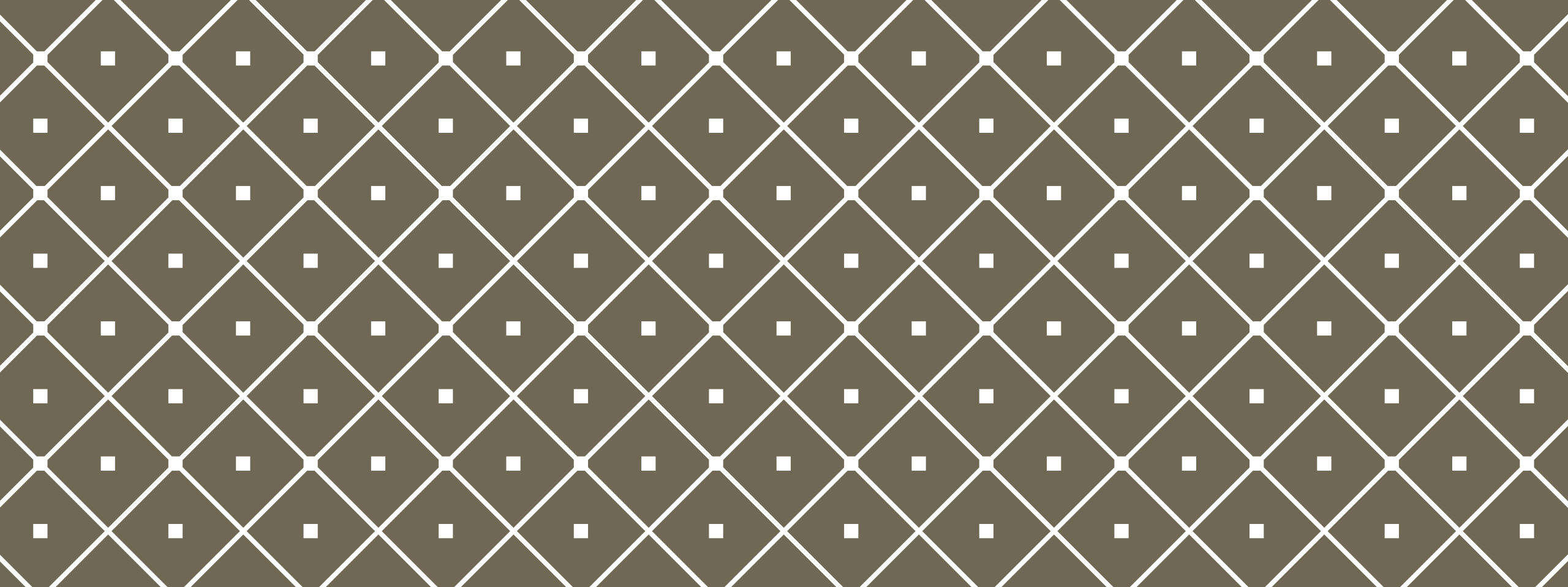


ALGORITHM ANALYSIS TUTORING

Tutor
신용호
컴퓨터과학과 12

Today?

1. 간단한 자기 소개
2. 한 학기 동안 배울 것?
3. 튜터링 방향 논의
4. Java Basic (듣고 싶은 사람만)



ICE BREAKING

간단한 자기소개 시간입니다.

안녕하세요



이름: 신용호

별명: 가젤, 갓드래곤타이거, 뱃용, ...

전화: 010-4277-3971

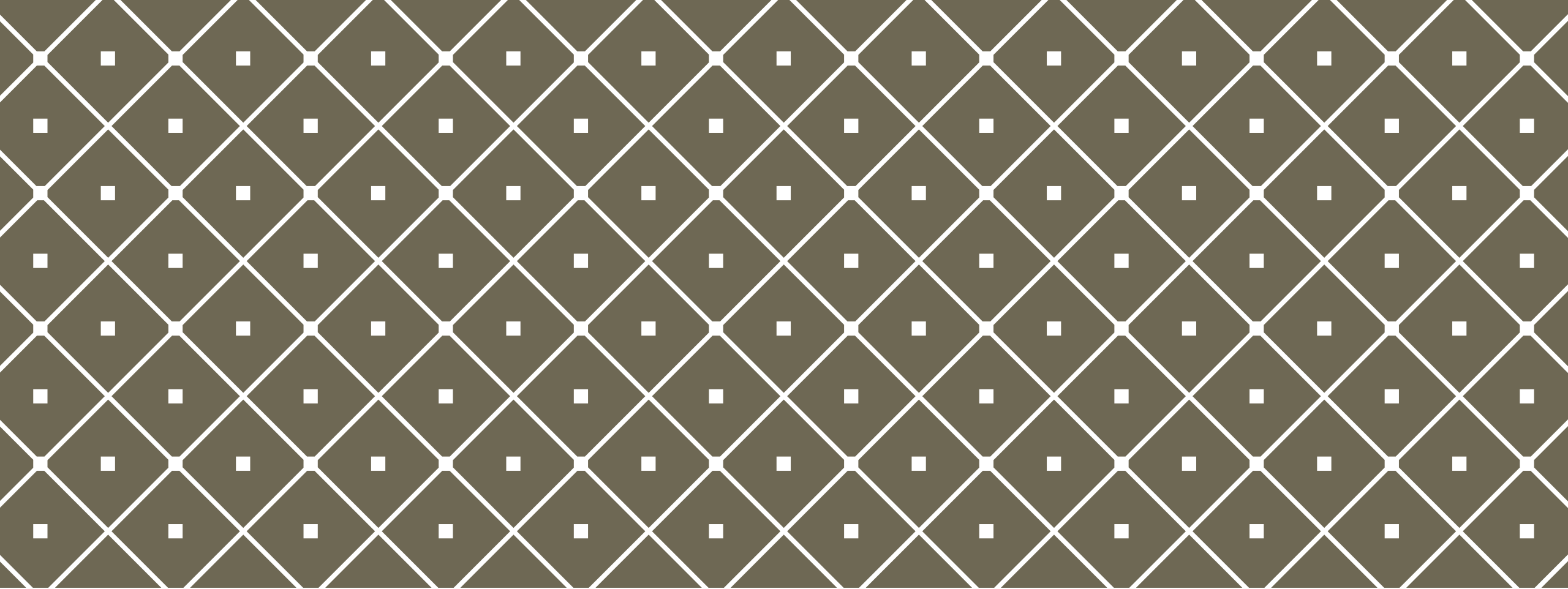
이메일: syh5477@gmail.com

...

딱히 뭐 소개할 게 없네요...

그냥 넣었어요...

잘 부탁드립니다.



WHAT TO LEARN?

한 학기 동안 배울 것을 알아보시다.

To Midterm

Algorithms with Numbers

Divide-and-Conquer Algorithms

Basic of Graph

Greedy Algorithms

Dynamic Programming

To Final

Network Flow



```
graph TD; A[Network Flow] --> B[Linear Programming]; B --> C["P, NP, NP-complete"]; C --> D["Coping with NP-completeness"]
```

Linear Programming

P, NP, NP-complete

Coping with NP-completeness

Algorithms with Numbers

Euclidean Algorithm

- Extended Euclidean Algorithm★

Modular Arithmetic

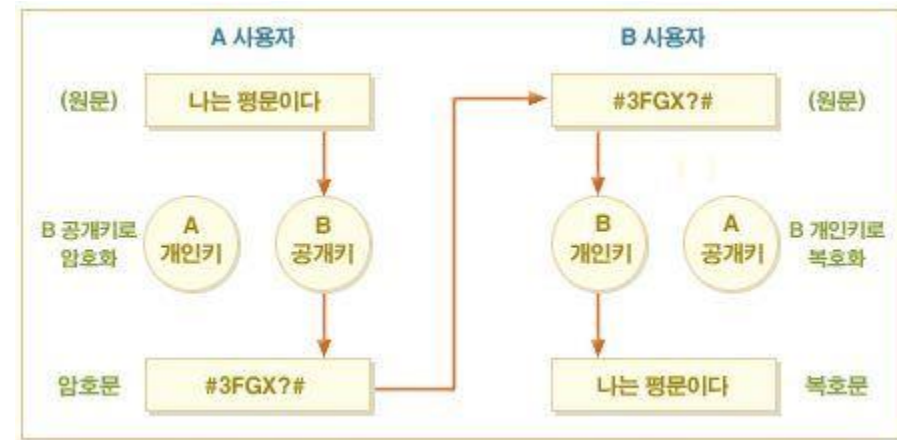
- Modular Exponentiation
- Modular Division★

Primality Test

- Fermat's Little Theorem★

RSA★

Universal Hashing



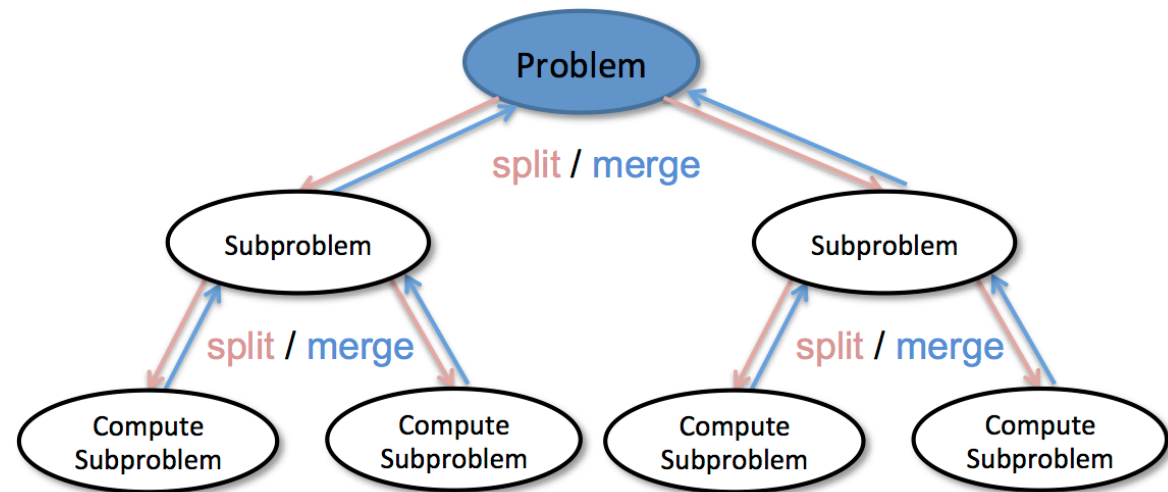
Divide-and-Conquer Algorithms

Master Theorem★

Randomized Selection Algorithm

Polynomials Multiplication

- Evaluation and Interpolation Method
- Roots of Unity
- Fast Fourier Transform★



Basic of Graph

Basic Definitions

- Undirected and Directed Graph
- Connected Component
- Directed Acyclic Graph
- Strongly Connected Component

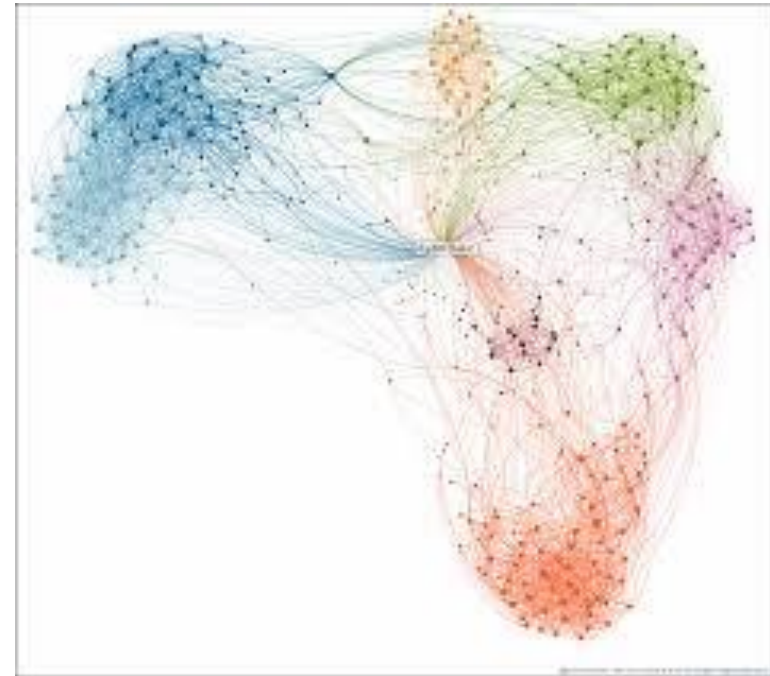
Depth-first Search

Decomposition of Directed Graph

Breath-first Search

Single Source Shortest Path Algorithms★

- Dijkstra Algorithm
- Bellman-Ford Algorithm



Greedy Algorithms

Minimum Spanning Tree Algorithms

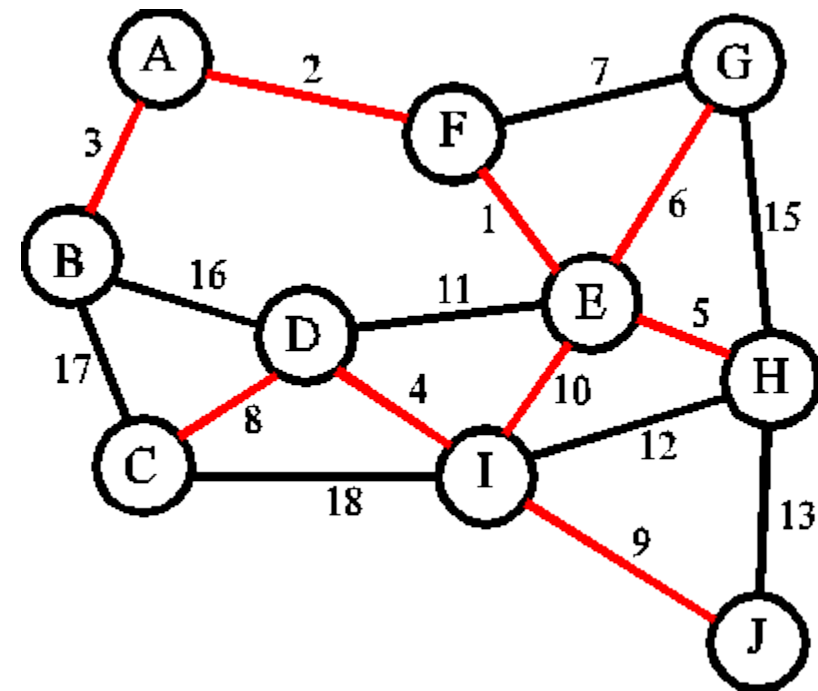
- Cut Property ★
- Kruskal Algorithm
 - Amortized Analysis ★
- Prim's Algorithm
- Randomized Min-Cut Algorithm

Greedy Set Cover ★

Task Scheduling

Huffman Encoding

Horn Formula



Dynamic Programming

Components of Dynamic Programming ★

- Problem, Sub-problem, Choice
- Implicit DAG

Edit Distance

Knapsack with/without Repetition

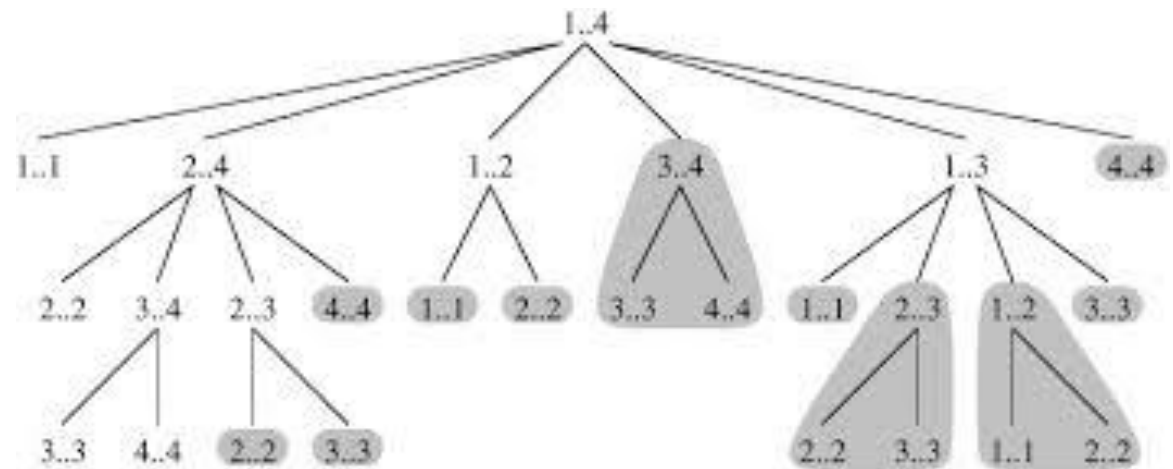
Chained Matrix Multiplication

- CYK Algorithm

Shortest Reliable Path

Floyd-Warshall Algorithm

Traveling Salesman Problem

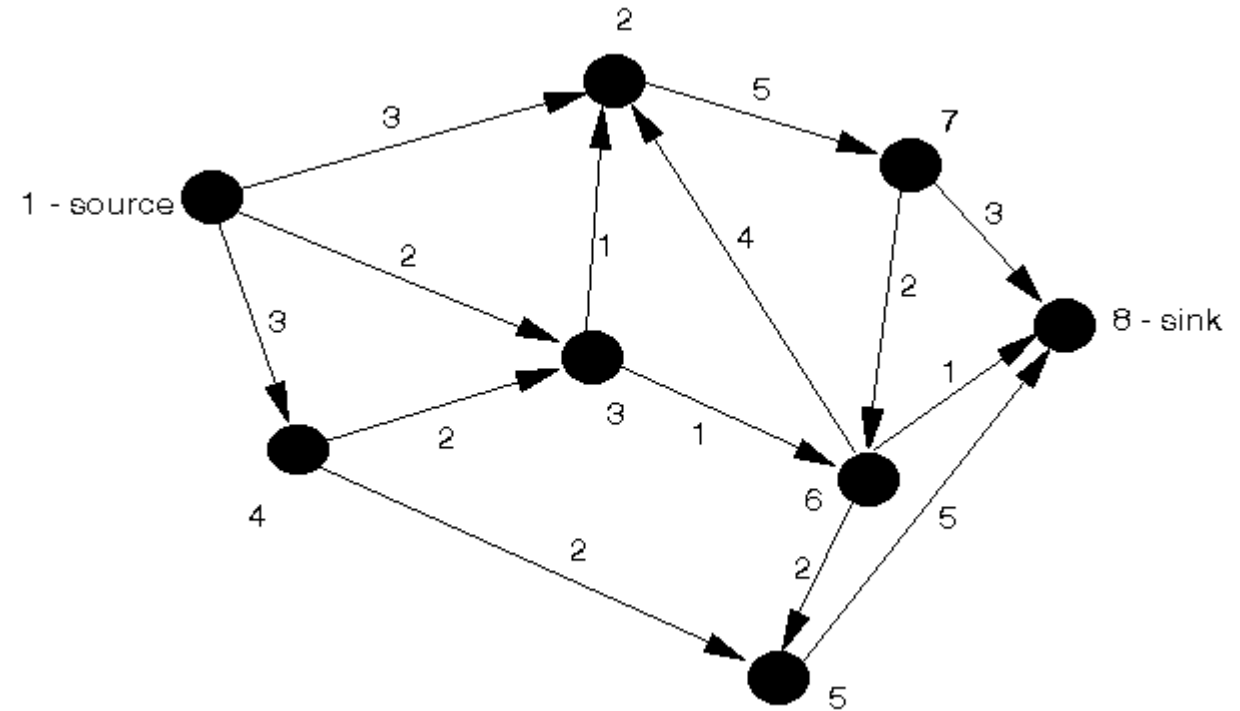


Network Flow

Ford-Fulkerson Method ★

- Flow Network, Capacity, Flow
- Residual Capacity, Residual Edge, Residual Graph
- Augmentation: Augmenting Flow, Augmenting Path
- Net Flow, Cut Capacity
- Max-Flow Min-Cut Theorem
- Edwards-Karp Algorithm
- Fattest Path Heuristic

Preflow-Push Algorithm



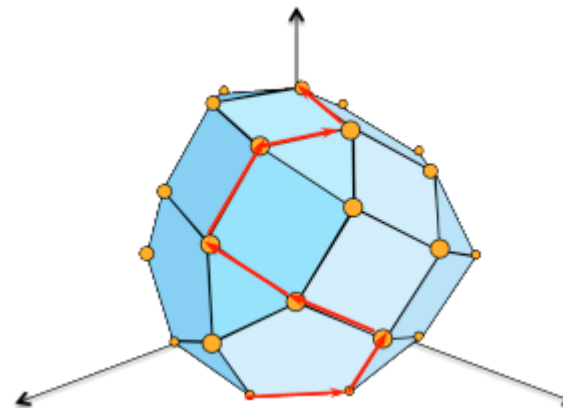
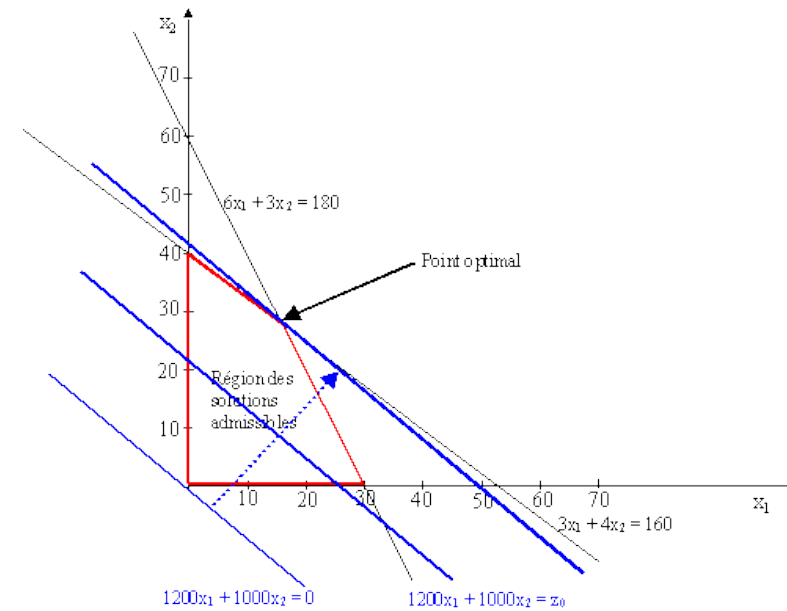
Linear Programming

Simplex Algorithm★

Duality Property★

Relationship with Network Flow

- Intuitive Approach of Ford-Fulkerson Method
- Max-Flow Min-Cut Duality Property



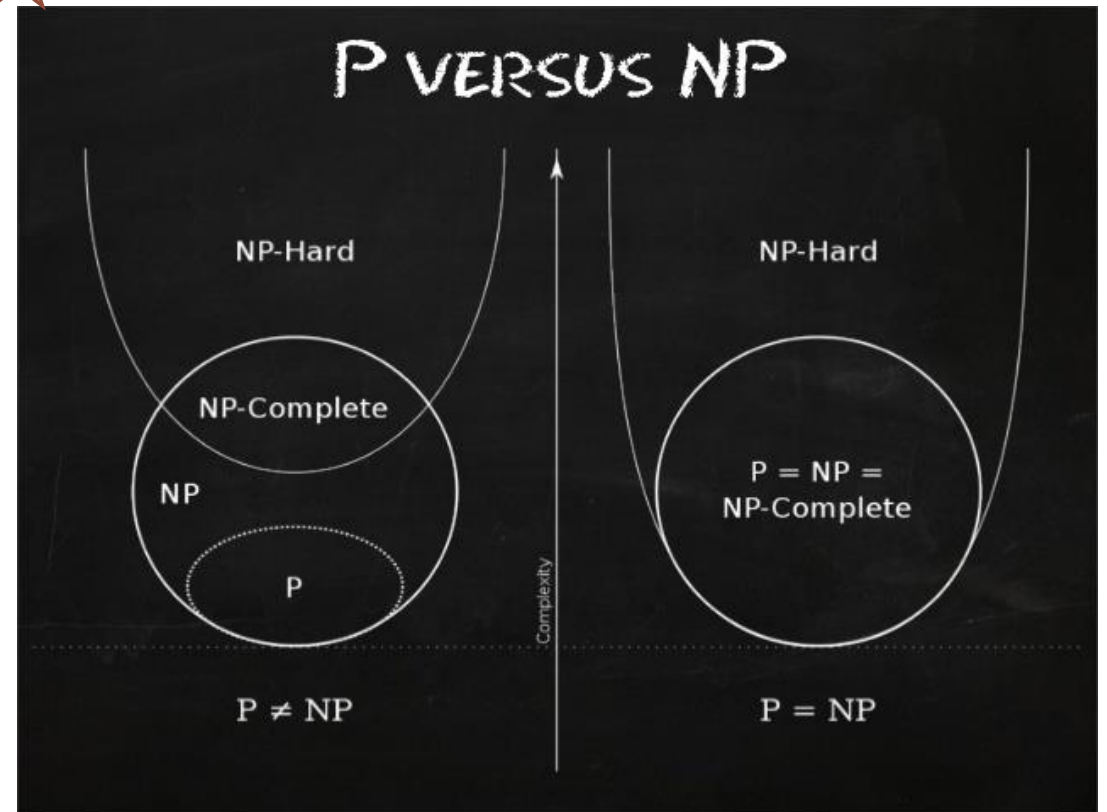
P, NP, NP-complete

Definition of class P, NP, NP-hard, NP-complete ★

- P: Optimization Problem and Decision Problem
- NP: Verification(Search) Problem
- NP-hard: Polynomial-Time Reducibility
- NP-complete: NP and NP-hard

Known NP-complete Problems ★

- Any NP \rightarrow CIRCUIT-SAT
- 3SAT \rightarrow INDEP-SET
- INDEP-SET \rightarrow CLIQUE, VERTEX-COVER
- 3D-MATCH \rightarrow ZOE
- ZOE \rightarrow SUBSET-SUM
- HAM-CYCLE \rightarrow TSP



Coping with NP-completeness

Intelligent Search★

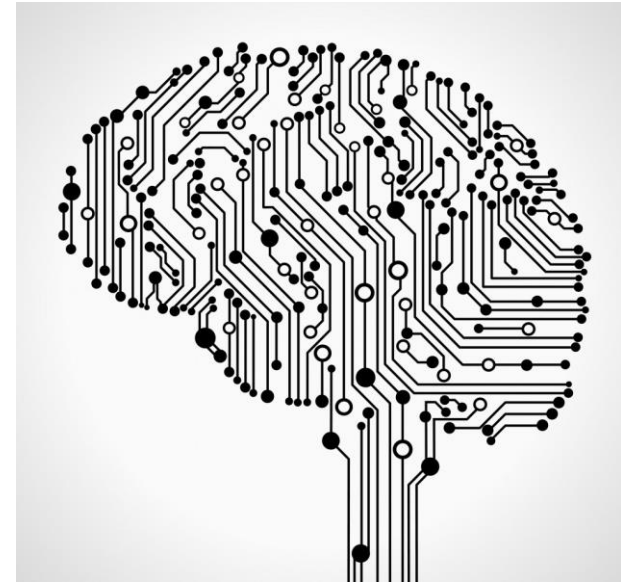
- Backtracking(Pruning)
- Branch-and-Bound

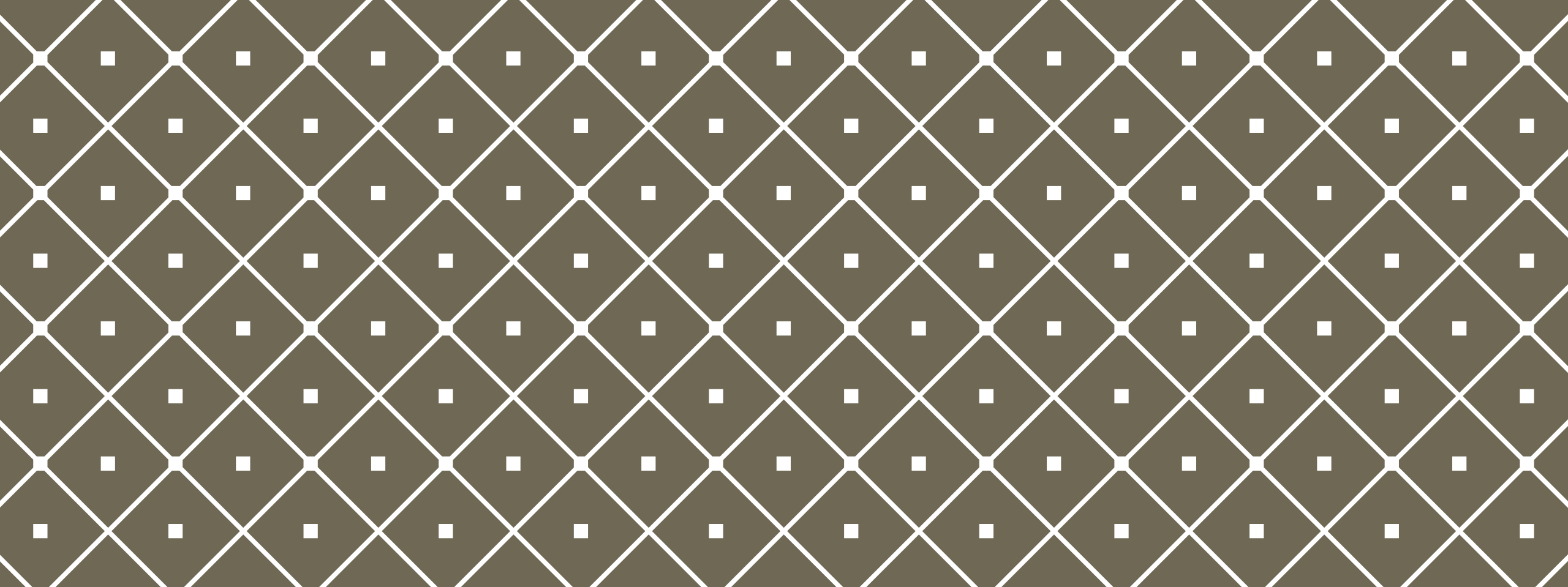
Local Search Heuristics

- Simulated Annealing
- Genetic Algorithms

Approximation Algorithms★

- Vertex Covering by Maximal Matching
- Clustering by Farthest-first Choosing
- TSP by MST
- Knapsack by Rescaling

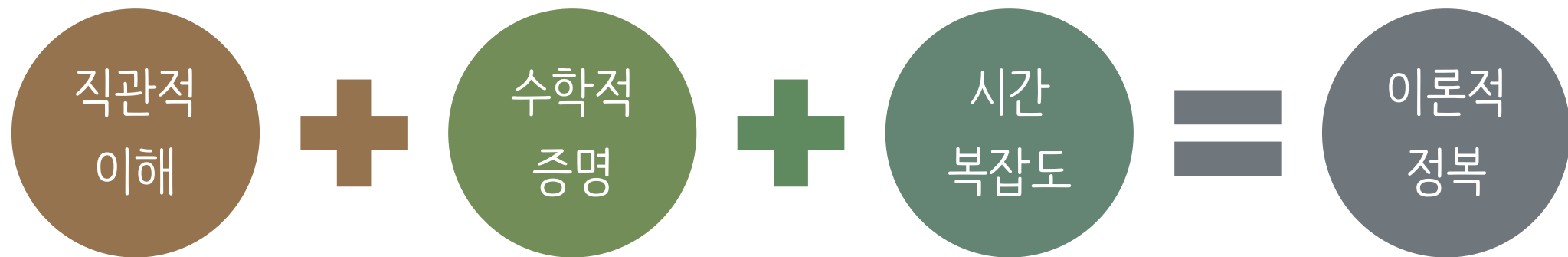




NOTICE & DISCUSSION

공지 및 논의사항

수업 방식



이학과 공학의 경계

이학적 특징

- 알고리즘이 정말로 정확한가? (Correctness)
- 알고리즘이 얼마나 효율적으로 동작하는가? (Efficiency)
 - 최악의 경우에는 어떻게 동작하는가? (Worst-case Analysis, Upper Bound)
 - 최선의 경우에는 어떻게 동작하는가? (Best-case Analysis, Lower Bound)
 - 평균적인 경우에는 어떻게 동작하는가? (Average-case Analysis, Randomized Input)
- 전체적인 동작 시간 속에서 단위 동작은 어떤가? (Amortized Analysis)

공학적 특징

- 알고리즘을 어떻게 구현할 것인가? (Implementation)
- 단위 동작 시간을 어떻게 줄일까? 단위 동작의 횟수를 어떻게 줄일까? (Hueristics)

제 튜터링은...

양성봉 교수님의 순서를 따릅니다.

알고리즘의 직관적인 이해를 가장 우선시 합니다.

엄밀한 수학적 증명도 함께 다룹니다.

알고리즘의 구현은 다루지 않습니다.

여러분의 적극적인 참여가 더욱 의미 있는 튜터링을 만듭니다

교재

튜터링을 판서로 진행할 예정입니다.

필요한 사항을 노트나 PPT에 직접 필기하시기 바랍니다.

교수님 강의 PPT 자료는 올라오는 대로 제게 가져다 주시기 바랍니다.

제가 주로 참조하는 교재는 다음과 같습니다.

- 『Introduction to Algorithms 3e』 written by Cormen, Leiserson, Rivest, Stein The MIT press
- 『Algorithms』 written by Dasgupta, Papadimitriou, Vazirani McGraw Hill Education

예상 일정 (중간)

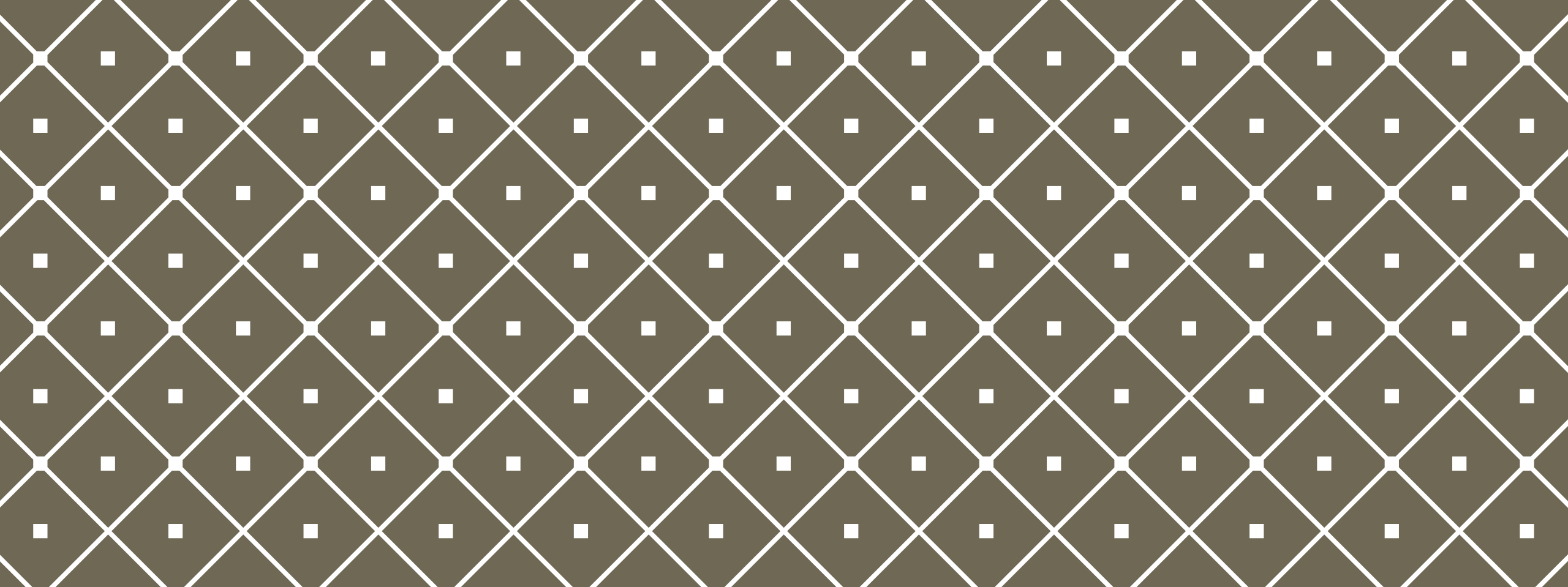
회차	예정일	튜터링 범위	특이사항
0	9/7	OT, Java Basic	
1	9/14	Nums, DaC	추석(9/14 ~ 9/16)
2	9/21	DaC, Graph	
3	9/28	Graph, Greedy	
4	10/5	Greedy, DP	
5	10/12	DP, 총정리	양고분 중간 예정(10/12)

예시상 일지정 (기말날)

회차	예정일	튜터링 범위	특이사항
6	11/2	Flow	
7	11/9	Flow, LP	
8	11/16	NPC	
9	11/23	NPC	
10	11/30	Coping with NPC	
11	12/7	총정리	

Q&A

기타 궁금한 점이나 나누고 싶은 의견이 있으시면 얘기해 주세요!



BASIC OF JAVA

Java 언어 기초를 배웁니다.

소스코드

여기서 다운받으세요!

<https://github.com/syh54772/AlgorithmAnalysisTutoring/>