4190.103A-001: Programming Practice Lab. 8 Assignment Tips

Euidong Lee
Junseok Lee
Minhyo Jeong
(snucsl.ta@gmail.com)

Systems Software & Architecture Lab.

Seoul National University

Spring 2022

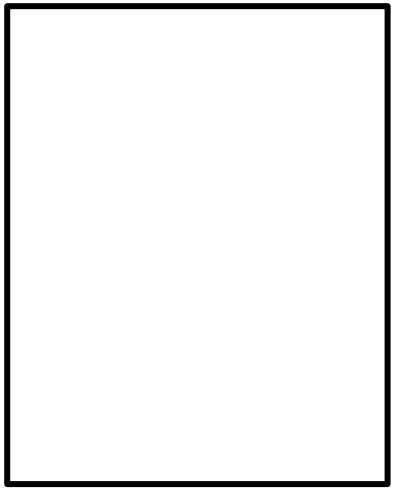
과제 1 - 설명

과제1 - 설명

- 아이디어를 얻기 위해 문제를 자세히 살펴봅시다!
- 괄호 문자열이 만족하는지 확인하기 위해서는 모든 열린 괄호의 짝이 있어야 합니다
- 그렇다면, 여는 괄호의 맞는 짝이 누구인지는 어떻게 알 수 있을까요??

과제1 - 설명

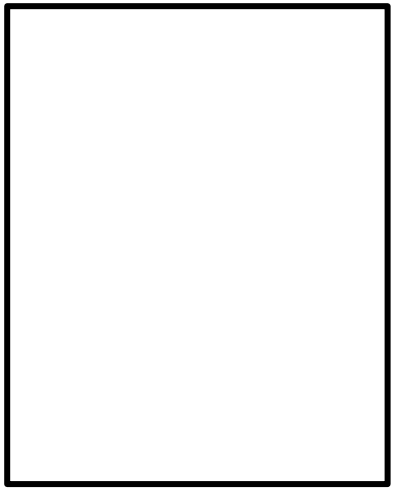
- 다음의 괄호 문자열을 판단하는 과정을 살펴봅시다!



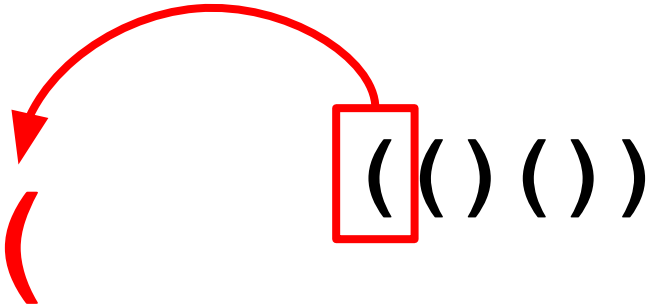
짝을 찾는 친구들

(( ) ( ))

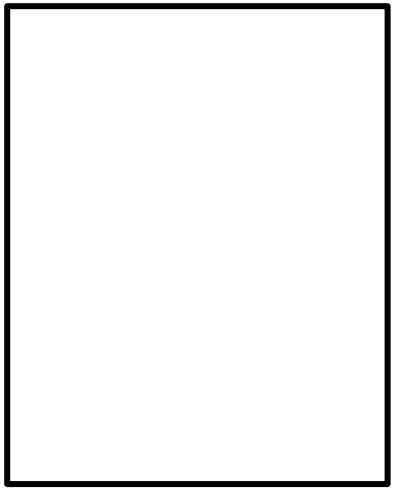
과제1 - 설명



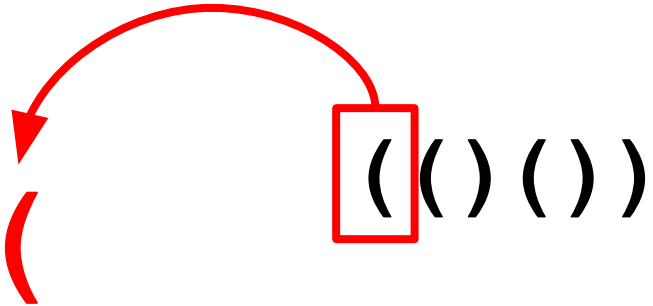
짝을 찾는 친구들



과제1 - 설명

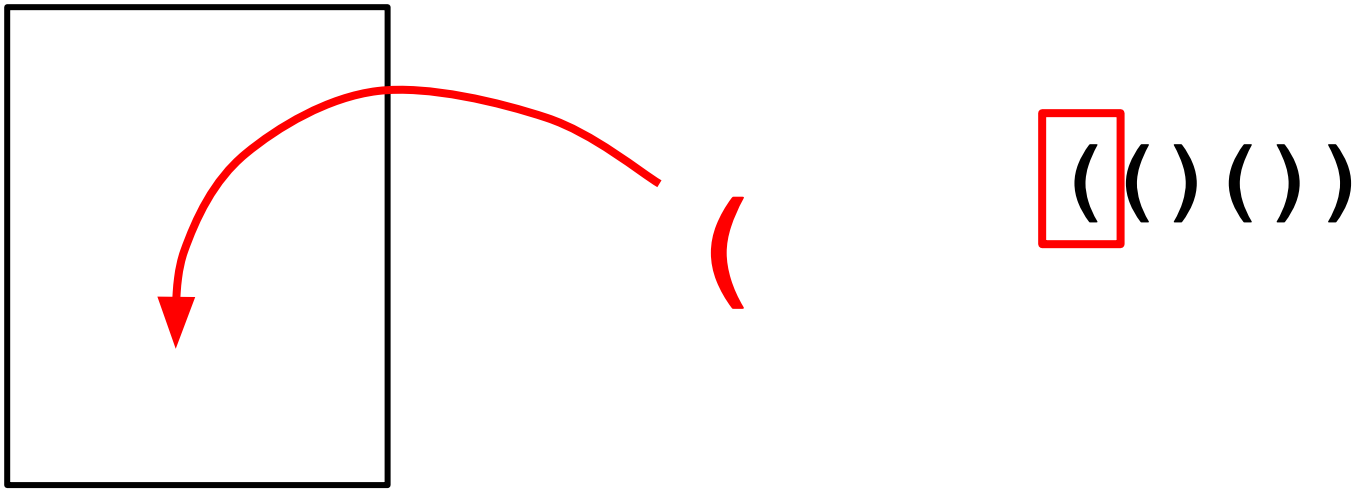


짝을 찾는 친구들

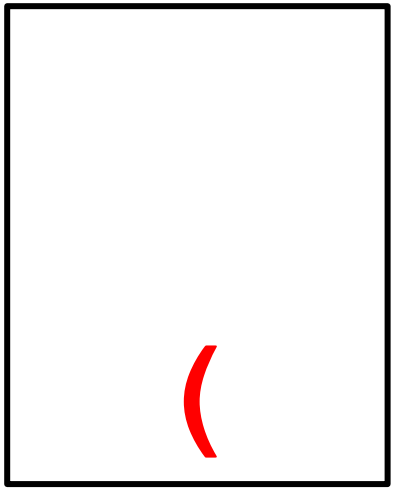


아직 짝이 없네?

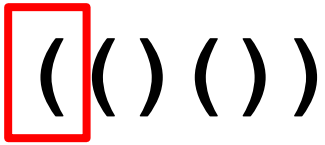
과제1 - 설명



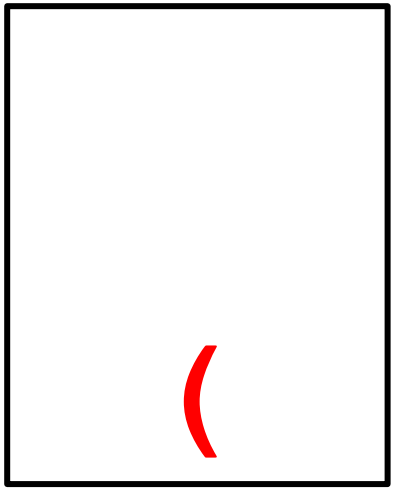
과제1 - 설명



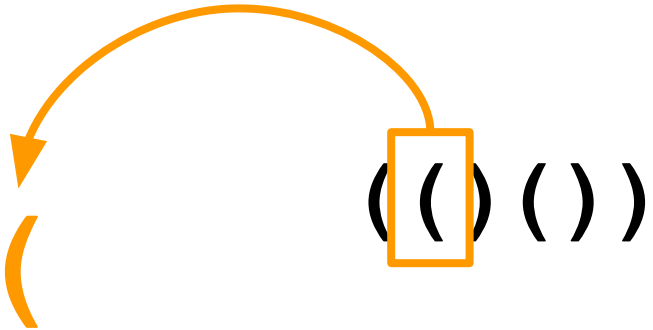
짝을 찾는 친구들



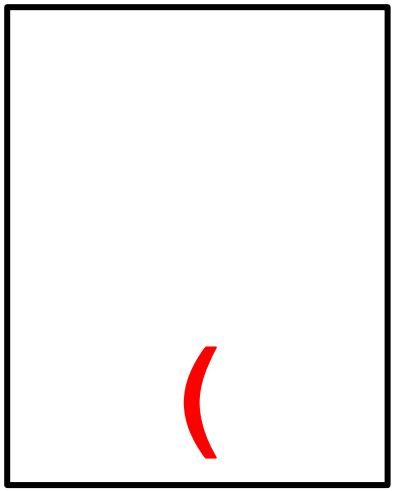
과제1 - 설명



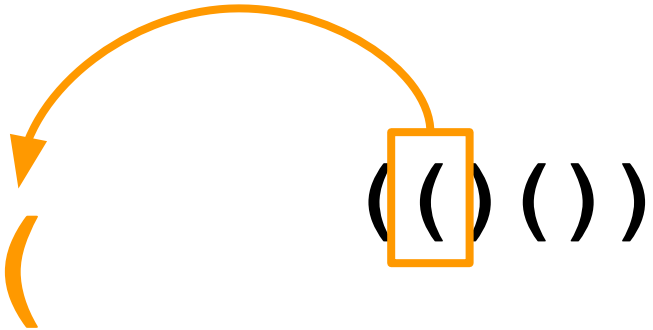
짝을 찾는 친구들



과제1 - 설명

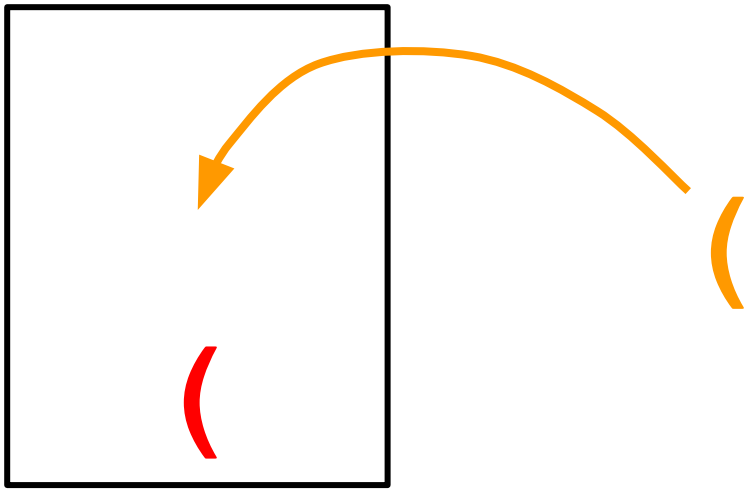


짝을 찾는 친구들

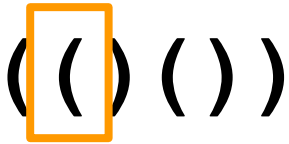


아직 짝이 없네?

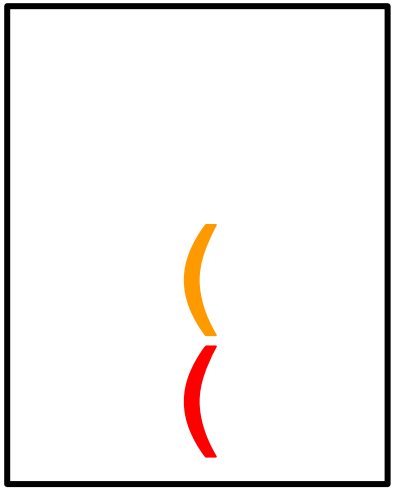
과제1 - 설명



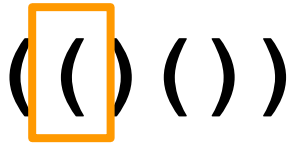
짝을 찾는 친구들



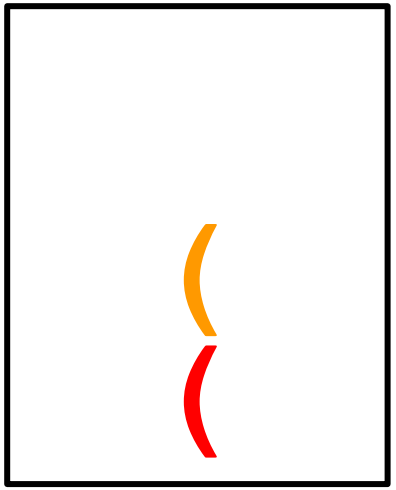
과제1 - 설명



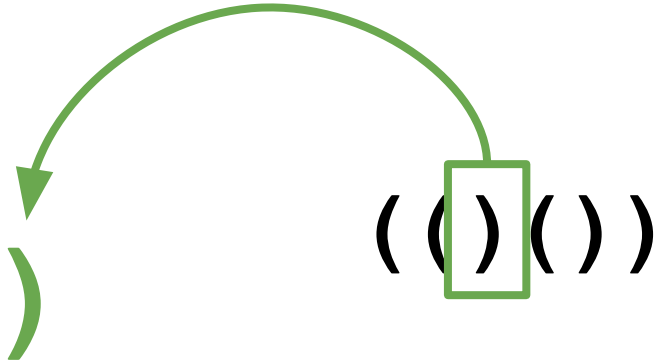
짝을 찾는 친구들



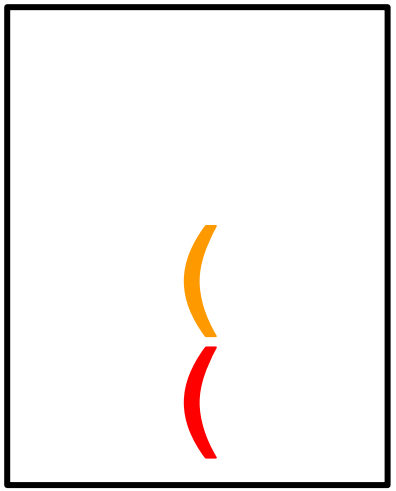
과제1 - 설명



짝을 찾는 친구들



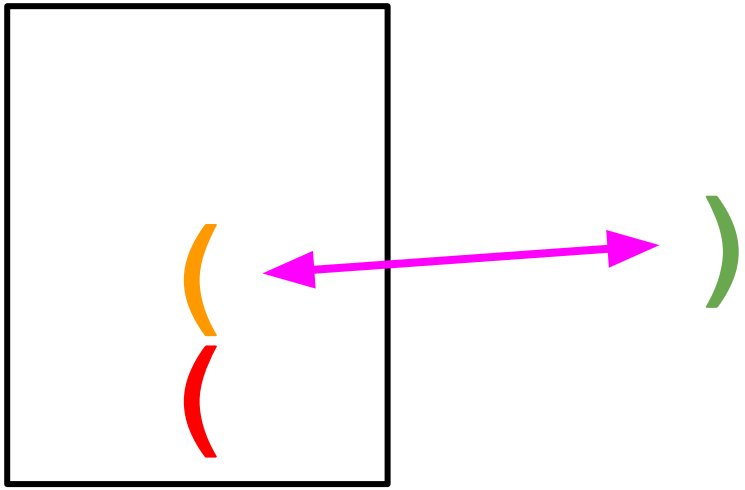
과제1 - 설명



짝을 찾는 친구들

날 기다리는
친구가 있나??

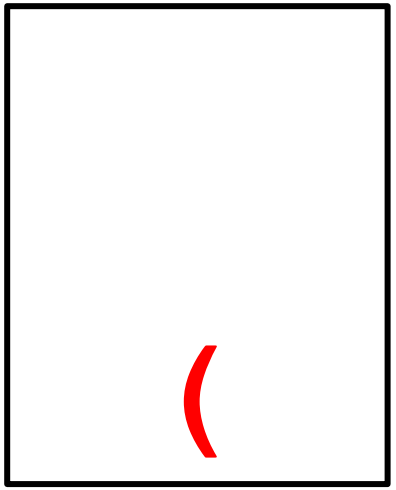
과제1 - 설명



짝을 찾는 친구들

(( ) ( ))

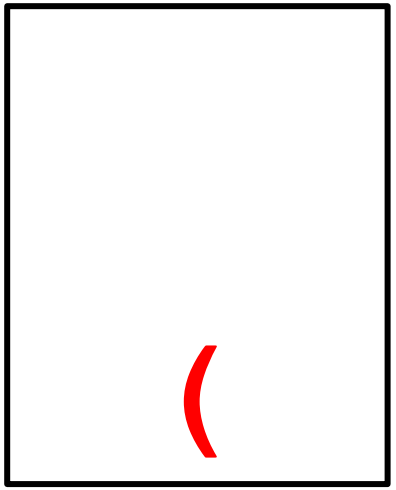
과제1 - 설명



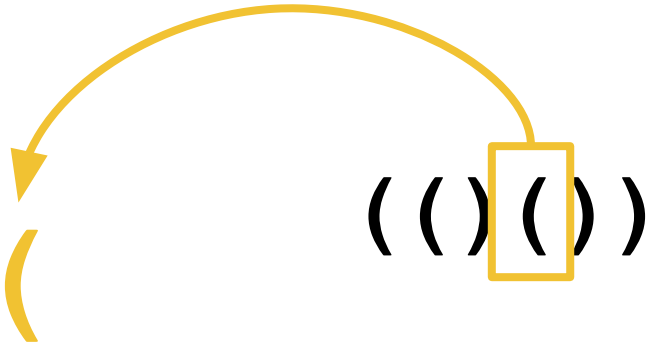
짝을 찾는 친구들

(( ) ( ))

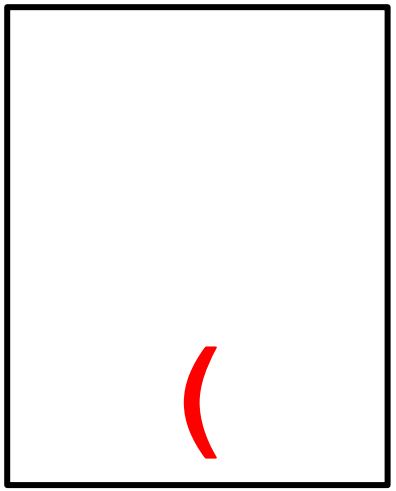
과제1 - 설명



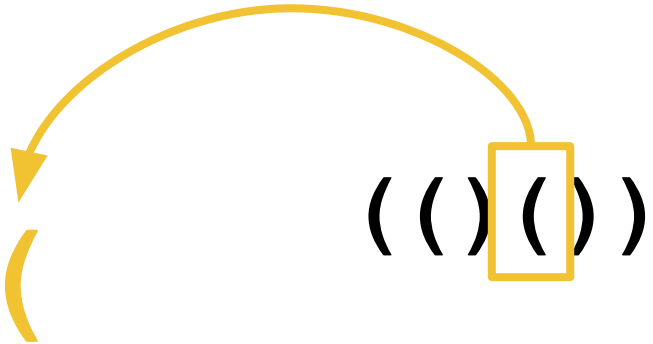
짝을 찾는 친구들



과제1 - 설명

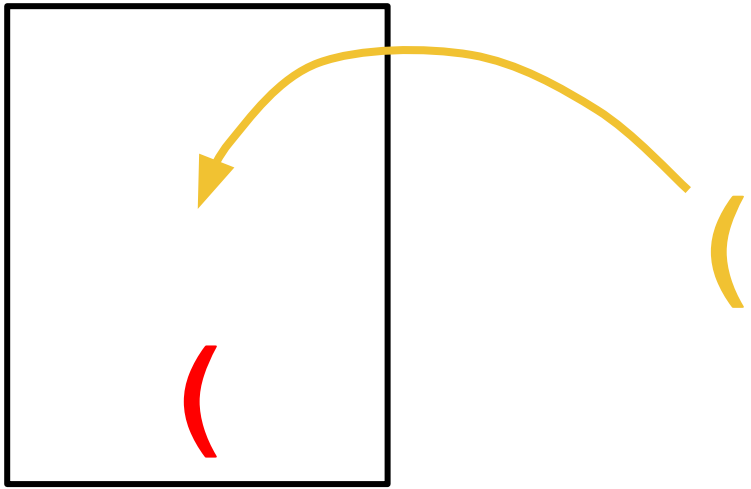


짝을 찾는 친구들



아직 짝이 없네?

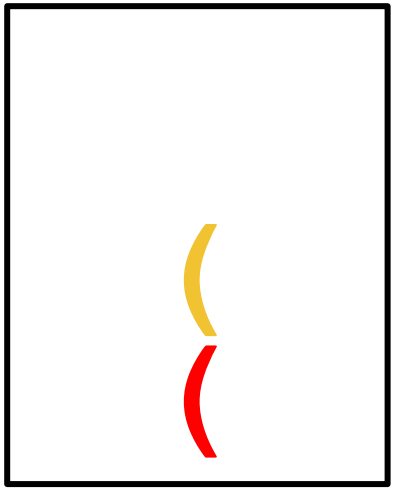
과제1 - 설명



짝을 찾는 친구들

(( ) ( ))

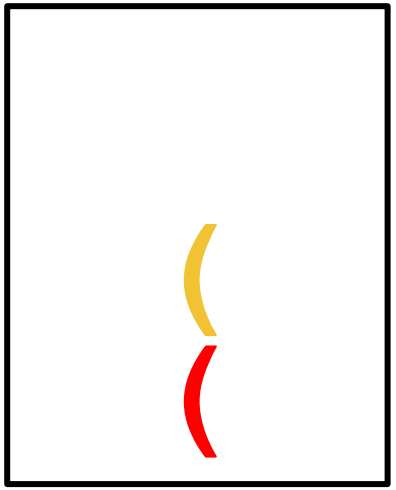
과제1 - 설명



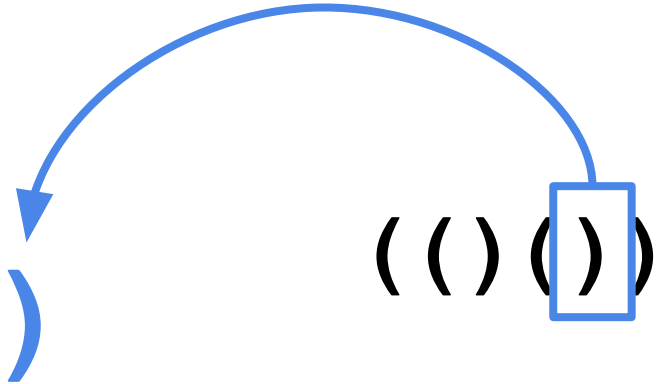
짝을 찾는 친구들

(( ) ( ))

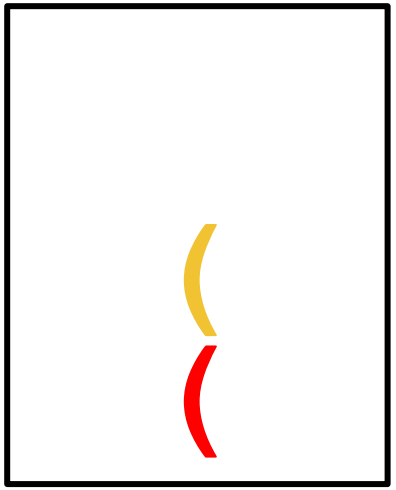
과제1 - 설명



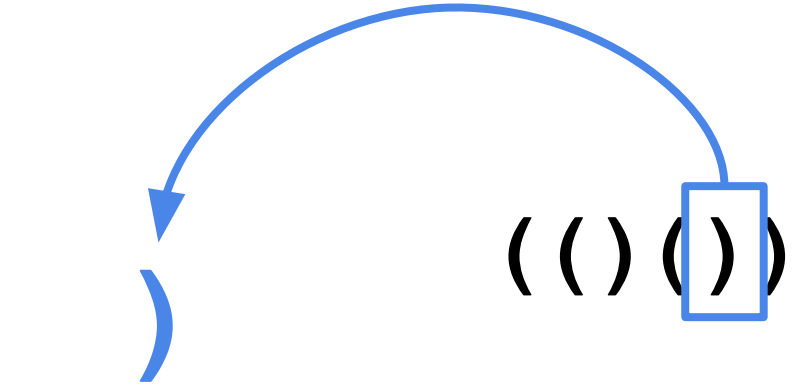
짝을 찾는 친구들



과제1 - 설명

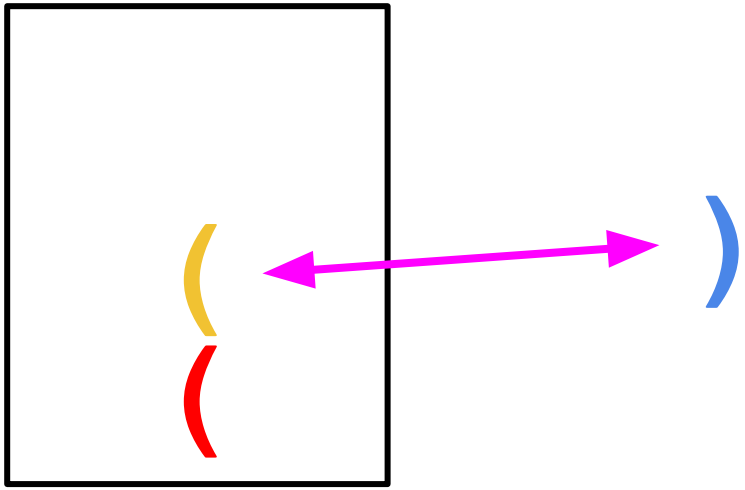


짝을 찾는 친구들



날 기다리는
친구가 있나??

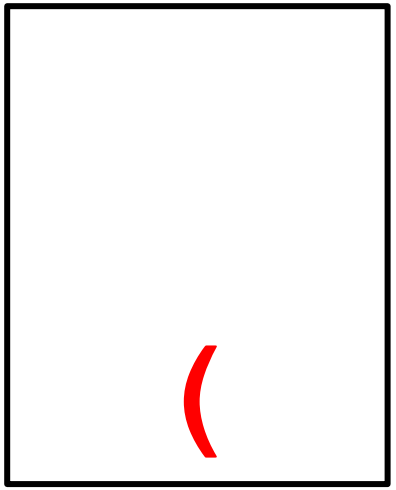
과제1 - 설명



짝을 찾는 친구들

(( ) ( ))

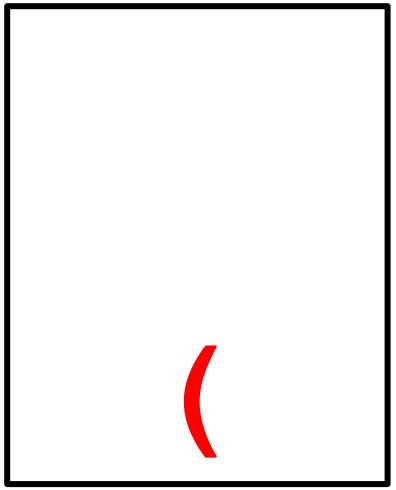
과제1 - 설명



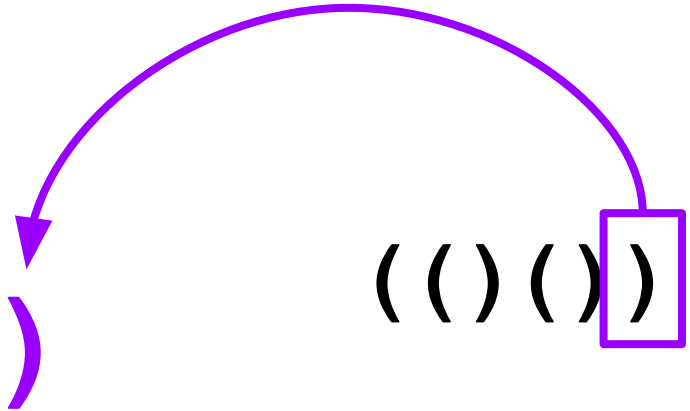
짝을 찾는 친구들

(( ) ( ))

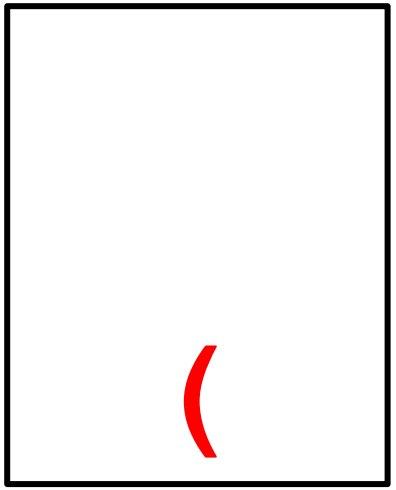
과제1 - 설명



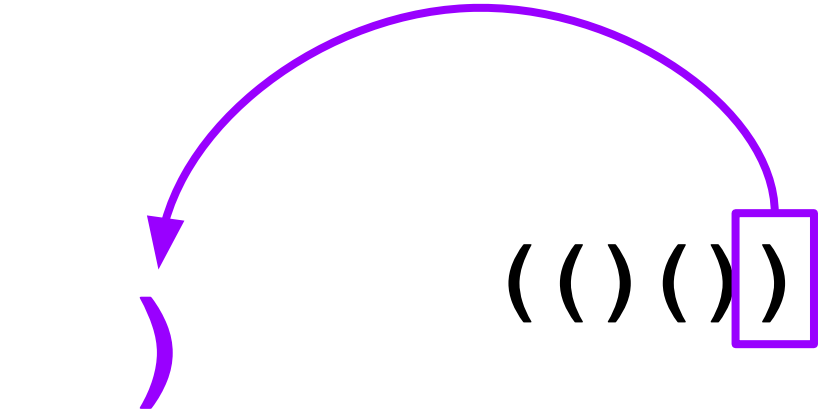
짝을 찾는 친구들



과제1 - 설명

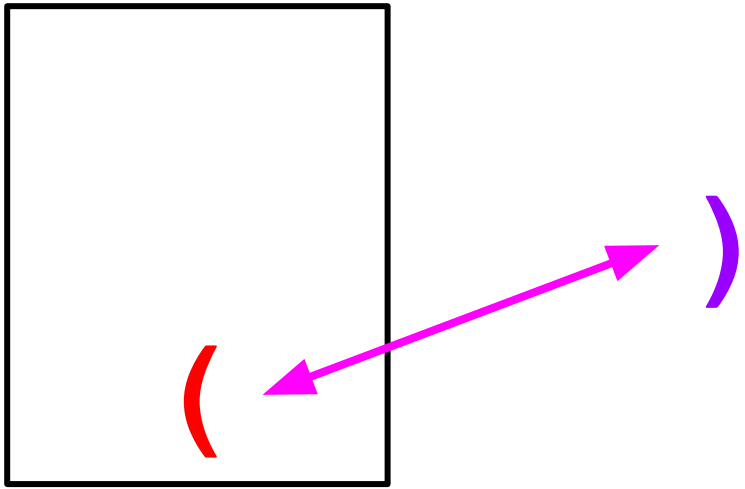


짝을 찾는 친구들



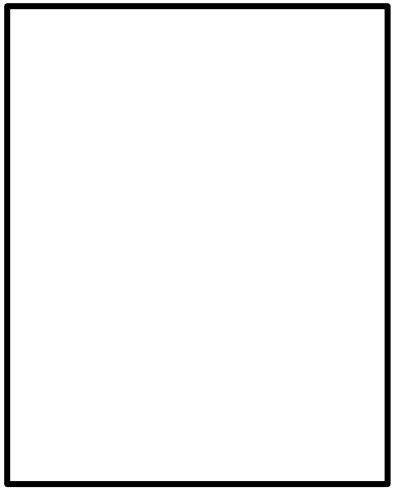
날 기다리는
친구가 있나??

과제1 - 설명



(( ) ( ))

과제1 - 설명



짝을 찾는 친구들

(( ) ( ))

과제1 - 설명

짝을 찾는 친구들

(( ) ( ))

과제1 - 아이디어

- 짝을 찾는 친구들을 저장하고, 짝을 지어주는 방식이 스택과 닮아있지 않나요??
- 고려해야할 부분
  - 모든 문자들이 짝을 찾았다는 것은 어떻게 알 수 있을까요??
  - 짝을 절대 찾지 못하는 문자가 있는 것은 어떻게 알 수 있을까요??

과제1 - Option

- 프로그램의 효율성을 위해서 다음을 고려해볼 수 있습니다!
- 본 문제는 stack을 활용하기 위한 배열이 사실 필요 없습니다
- Stack의 동작하는 방식만 참고하고, stack을 사용하지 않는다면, 배열 없이 변수 하나로 구현할 수 있습니다

과제 2 - 설명

과제2 - 설명

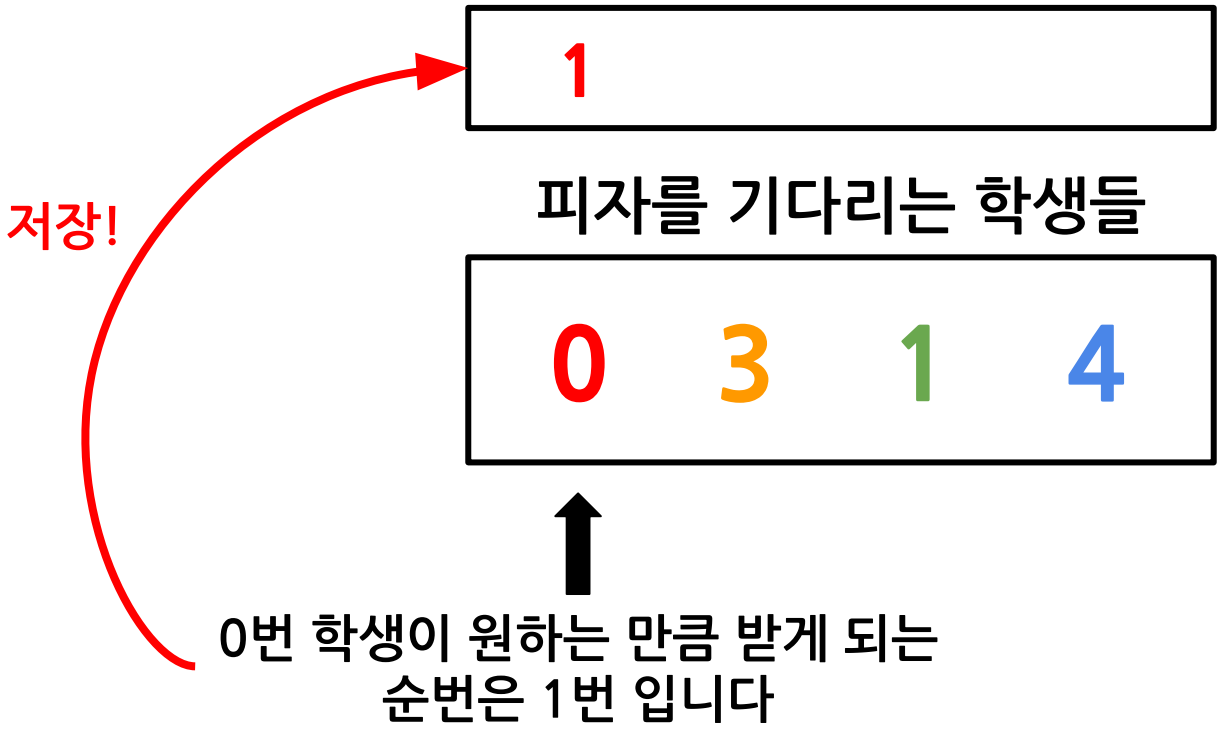
- 마찬가지로 문제 해결을 위한 아이디어를 얻기 위해 문제를 살펴봅시다!

피자를 기다리는 학생들

과제2 - 설명

과제2 - 설명

과제2 - 설명



과제2 - 설명

과제2 - 설명

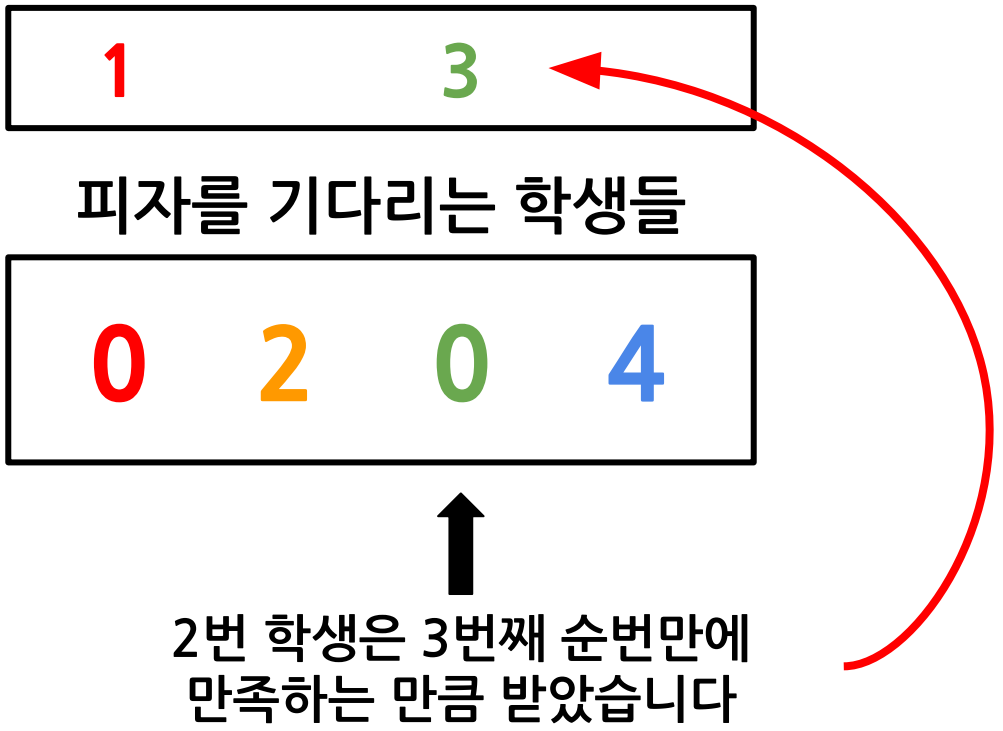
과제2 - 설명

과제2 - 설명

- 여기서 아이디어가 필요합니다!
- 1번 학생은 다시 뒤로 가서 줄을 서게 됩니다
- 줄의 순서가 달라지는 것 같지만 사실 변화는 없습니다
- 1번 학생이 다시 받게 되는 순간은 한바퀴를 다 돌고 온 후가 되기 때문입니다
- 따라서, 학생이 서 있는 순서를 바꾸는 것이 아니라, 나눠주는 화살표를 조정합니다
    - 화살표가 줄 끝까지 갔다면, 다시 줄의 맨 앞으로 가져옵니다

과제2 - 설명

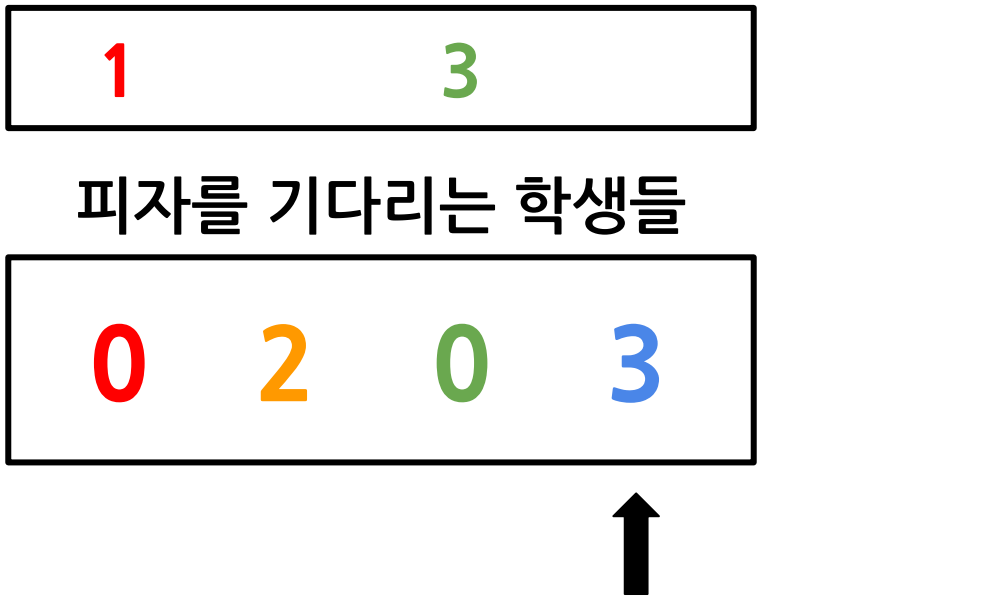
과제2 - 설명



과제2 - 설명

과제2 - 설명

과제2 - 설명



과제2 - 설명

과제2 - 설명

과제 - 아이디어

- 설명한 아이디어를 통해서 각 학생들이 만족하는 만큼 피자를 받게되는 순번을 알아낼 수 있습니다!
- 다음을 고려해보세요
  - 학생들에게 나눠주는 것은 언제까지 해야할까요?