**JAVA**

자료구조

* 선형자료구조 : 배열, 스택, 큐, 덱
* 비선형자료구조 : 그래프, 트리

정렬

* Comparable(원소 스스로 비교 가능)
* Comparator(제 3자 비교도우미)
* 프로그래머스 – 입출력코드 다 갖고 있음(입출력 최적화 불가)
* 그 외 – 입출력 최적화 가능

알고리즘

* 효율성
* **시간 복잡도 : 처음에 결정한 후 알고리즘 시작하기**
  + Int(N) = 4byte?
  + 1억번의 연산 = 1초
  + O(1)은 거의 없고, 수학적인 공식 등의 힘을 빌려야 한다.
  + O(N^2) -> O(NlogN)까지 만드는게 최적화
  + 재귀
    - 선형적인 재귀 : 자신으로부터 한번씩만 파생되는 경우 => 반복문이 더 효율적
    - 깊이가 깊어질수록 기하급수로 수행횟수가 늘어남 O(2^N) => memorization 기법활용해서 최적화하기
* 공간 복잡도

1. 완전탐색(BruteForce) : 반복재귀
2. 순열/조합/부분집합
   1. 순열 : n!
   2. 조합 : n!/(n-r)!\*r!
   3. 부분집합 : 2^n
3. 그리디
4. 그래프탐색
   1. DFS
      1. 깊이 먼저 탐색
      2. 재귀 이용(패턴 있음)
   2. BFS
      1. 거쳐가는 간선 = 너비
      2. 너비가 같은 애들을 먼저 탐색한다
5. 동적계획법(DP)
   1. 점화식

문제 풀이 : <https://solved.ac/>

입사 시 실버까지 필수, 골드4~3까지

환경설정

* JDK 다운로드 후 > 환경변수 설정 : (시스템변수:JAVA\_HOME : 위치) / (Path:%JAVA\_HOME%\bin) -> 추후 자바 버전 변경할 때는 시스템변수 위치만 바꿔주자

이클립스

* 원하는 폴더로 workspace 설정하기
* 자동완성 : Ctrl + space
* 실행 : Ctrl + F11
* 한꺼번에 rename : Alt Shift R

프로그램 실행

* 부하가 큰 것 : 입출력, 객체생성 등
  + 입출력을 모아서 출력하는 것 : StringBuilder

실제 문제 풀이 – 사이트에 적용 시

\*\*\*\* 백준 제출 시 유의사항 \*\*\*

package는 제외해야 함

import 전부 넣어야 함

class 이름을 Main으로 제출해야 함

큐

* Linked list
* Offer() : 넣는 함수 / enque
* Poll() : 빼는 함수 / deque
* peek : 맨 앞이 무엇인지 확인
* isEmpty() : 비어있는지 여부
* size()