* 복잡도 분석
* 수학을 활용한 문제풀이
* 규칙 찾기
  + <http://usaco.org/index.php?page=viewproblem2&cpid=667> (2016 Dec S 2번)
  + <http://usaco.org/index.php?page=viewproblem2&cpid=620> (2016 Feb S 3번)
  + <http://usaco.org/index.php?page=viewproblem2&cpid=739> (2017 US S 2번)
  + <http://usaco.org/index.php?page=viewproblem2&cpid=990> (2020 Jan S 1번)
  + <https://www.acmicpc.net/problem/13305> (2016 KOI 중등부 2번 : 주유소)
* 재귀적 풀이 (관계 중심적 사고)
  + 리니어 서치, 바이너리 서치
  + Binary Search 활용
    - <http://usaco.org/index.php?page=viewproblem2&cpid=765> (2017 Dec G 1번) :: 어려운 버전
    - <http://usaco.org/index.php?page=viewproblem2&cpid=858> (2018 Dec S 1번)
    - <http://usaco.org/index.php?page=viewproblem2&cpid=715> (2017 Feb S 2번) :: DP로도 풀 수 있음
    - <http://usaco.org/index.php?page=viewproblem2&cpid=690> (2017 Jan S 1번) :: check 과정에서 Priority Queue 사용
    - <http://usaco.org/index.php?page=viewproblem2&cpid=991> (2020 Jan S 2번) :: 수학적 개념 조금 사용
    - <http://usaco.org/index.php?page=viewproblem2&cpid=992> (2020 Jan S 3번)
  + 팩토리얼, 피보나치
  + Palindrome 판별, AnBn 판별
  + 조합 (Combination), S(n,k), P(n,k)
  + Kth Smallest Element
  + 하노이 타워
  + N Queen 문제
  + 재귀함수 사용
    - <http://usaco.org/index.php?page=viewproblem2&cpid=692> (2017 Jan S 3번)
* 재귀 (다이나믹 프로그래밍)
  + 행렬 경로 문제
  + 돌 놓기
  + RGB 거리 <https://www.acmicpc.net/problem/1149>
  + 가장 긴 증가하는 부분수열 <https://www.acmicpc.net/problem/11053>
  + LCS <https://www.acmicpc.net/problem/9251>
  + <http://usaco.org/index.php?page=viewproblem2&cpid=863> (2018 Dec G 3번)
  + 이차원 공간에서의 DP (<http://usaco.org/index.php?page=viewproblem2&cpid=923>) & <http://usaco.org/index.php?page=viewproblem2&cpid=915> (2019 Feb S2번)
  + Knapsack <https://www.acmicpc.net/problem/12865> ( \*\* greedy version의 fraction knapsack도 소개해주자 : 규칙 찾기에 ?)
  + 어려운 DP
    - <https://www.acmicpc.net/problem/10835> (KOI 중등부 2015 3번)
    - <http://usaco.org/index.php?page=viewproblem2&cpid=897> (2019 Jan G 1번)
    - <http://usaco.org/index.php?page=viewproblem2&cpid=815> (2018 Feb G 3번)
    - <http://usaco.org/index.php?page=viewproblem2&cpid=791> (2018 Jan G 3번)
    - <http://usaco.org/index.php?page=viewproblem2&cpid=766> (2017 Dec G 2번)
    - <http://usaco.org/index.php?page=viewproblem2&cpid=718> (2017 Feb G 2번)
    - <http://usaco.org/index.php?page=viewproblem2&cpid=694> (2017 Jan G 1번)
    - <http://usaco.org/index.php?page=viewproblem2&cpid=670> (2016 Dec G 2번)
    - <http://usaco.org/index.php?page=viewproblem2&cpid=574> (2015 Dec G 2번) :: Knapsack의 아이디어 활용 문제 (0-1 knapsack)
* 자료구조를 활용
  + 링크드리스트, “벡터”의 사용
    - 링크드리스트 개념 활용 문제
      * <http://usaco.org/index.php?page=viewproblem2&cpid=813> (2018 Feb G 1번)
  + Stack, Queue의 활용
    - Infix to Postfix, Stack Calculator
    - Stack 활용 ( <http://usaco.org/index.php?page=viewproblem2&cpid=922>)
      * <https://www.acmicpc.net/problem/1874> (스택 수열)
      * <https://www.acmicpc.net/problem/4949> (균형잡힌 세상 -> 괄호 쌍 맞추기)
  + Tree (\*\* Tree의 심화 개념은 Graph의 연장선상에서 이해)
    - Binary Search Tree
    - C++ Map, Set의 사용
      * Set의 lower\_bound 사용
        + <http://usaco.org/index.php?page=viewproblem2&cpid=714> (2017 Feb S1)
      * Map을 활용하면 쉽게 풀림
        + <http://usaco.org/index.php?page=viewproblem2&cpid=862> (2018 Dec G 2번)
      * Balanced Binary Search Tree를 활용
        + <http://usaco.org/index.php?page=viewproblem2&cpid=767> (2017 Dec G3)
    - Tree의 재귀적인 구조와 DFS, BFS
    - Union-Find (Disjoint Set) -> Tree를 사용한 집합의 표현
      * <http://usaco.org/index.php?page=viewproblem2&cpid=789> (2018 Jan G 1번) :: 2018 Jan S 3번 http://usaco.org/index.php?page=viewproblem2&cpid=788 (BFS 활용 풀이)의 어려운 버전
      * <http://usaco.org/index.php?page=viewproblem2&cpid=669> (2016 Dec G 1번)
  + Heap의 사용
    - <http://usaco.org/index.php?page=viewproblem2&cpid=859> (2018 Dec S 2번) : Priority Queue 사용 ?
  + Hash Table ?, Hash Map
* 정렬
  + 정렬을 하는 방법
    - Bubble, Insertion, Merge, Heap, Quick(Partition!), Radix
    - Insertion Sort의 개념을 활용
      * <http://usaco.org/index.php?page=viewproblem2&cpid=898> (2019 Jan G 2번)
  + 정렬을 활용한 문제풀이
    - <http://usaco.org/index.php?page=viewproblem2&cpid=896> (2019 Jan S 3번) : 정렬을 해서 문제를 쉽게 품
    - <http://usaco.org/index.php?page=viewproblem2&cpid=786> (2018 Jan S 1번)
    - <http://usaco.org/index.php?page=viewproblem2&cpid=787> (2018 Jan S 2번)
    - <http://usaco.org/index.php?page=viewproblem2&cpid=573> (2015 Dec G 1번)
* 가장 중요한 그래프
  + 그래프를 만드는 방법 & 그래프의 정의, 종류
  + 문제를 그래프로 보기
    - <http://usaco.org/index.php?page=viewproblem2&cpid=894> (2019 Jan S 1번)
  + DFS, BFS
    - 길찾기, 2차원 좌표에서 DFS 사용
      * <http://usaco.org/index.php?page=viewproblem2&cpid=860> (2018 Dec S 3번)
    - <http://usaco.org/index.php?page=viewproblem2&cpid=788> (BFS, 2018 Jan S 3번)
    - <http://usaco.org/index.php?page=viewproblem2&cpid=716> (2017 Feb S 3번)
    - <http://usaco.org/index.php?page=viewproblem2&cpid=668> (2016 Dec S 3번)
    - Flood Fill
      * <http://usaco.org/index.php?page=viewproblem2&cpid=944> (2019 US S 3번)
      * <http://usaco.org/index.php?page=viewproblem2&cpid=920> (2019 Feb S 3번)
      * <http://usaco.org/index.php?page=viewproblem2&cpid=895> (2019 Jan S 2번)
  + DFS, BFS 활용
    - DFS 활용 <http://usaco.org/index.php?page=viewproblem2&cpid=814> (2018 Feb G 2)
    - <http://usaco.org/index.php?page=viewproblem2&cpid=790> (2018 Jan G 2)
    - BFS로 최단거리 : Vertex의 의미를 확장
      * <http://usaco.org/index.php?page=viewproblem2&cpid=695> (2017 G 3번)
      * <http://usaco.org/index.php?page=viewproblem2&cpid=575> (2015 G 3번)
  + Minimum Spanning Tree
    - Prim
  + \*\*Tree
    - Binary Indexed Tree
      * <http://usaco.org/index.php?page=viewproblem2&cpid=719> (2017 Feb G 3번)
      * <http://usaco.org/index.php?page=viewproblem2&cpid=693> (2017 Jan G 1번 : 귀찮음)
    - LCA (Lowest Common Ancestor)
      * 2019 February 1
  + Shortest Path
    - Dijkstra
      * <http://usaco.org/index.php?page=viewproblem2&cpid=899> (2019 Jan G 3번)
      * <http://usaco.org/index.php?page=viewproblem2&cpid=861> (2018 Dec G 1번)
      * <http://usaco.org/index.php?page=viewproblem2&cpid=717> (2017 Feb G 1번)
      * <http://usaco.org/index.php?page=viewproblem2&cpid=671> (2016 Dec G 3번)
    - Topological Sorting
    - BFS 이용
    - 벨만-포드
    - 플로이드 워셜
* 다른 다양한 알고리즘
  + 문자열 매칭
  + 프림 알고리즘
  + 분할 정복
  + 백 트래킹
  + 정수론 (유클리드 호제법, 피보나치, 등등)
  + 트라이
  + 네트워크 유량