

구분	타이틀	풀이방법
Hash	완주하지못한선수, 위장, 전화번호목록, 베스트앨범(프로그래머스)	Hash 란 데이터를 다루는 기법 중 하나이고, Hash 의 핵심은 KEY 와 VALUE 가 한 쌍으로 이루어진 함수다.
Greedy	체육복, 구명보트, 단속카메라, 조이스틱, 최대최소값, 큰수만들기, 일이될때까지, 숫자카드게임, 큰수의법칙, 회의실배정(프로그래머스, 백준)	greedy(탐욕법) 알고리즘은 가장 직관적인 알고리즘 설계 패러다임 중 하나이다. 이는 우리가 원하는 답을 재귀 호출과 똑같이 여러 개의 조각으로 쪼개고, 각 단계마다 답의 한 부분을 만들어 간다는 점에서 완전 탐색이나 동적 계획법 알고리즘과 다를 것이 없다. 그러나 모든 선택지를 고려해 보고 그중 전체 답이 가장 좋은 것을 찾는 두 방법과는 달리, 탐욕법은 각 단계마다 지금 당장 가장 좋은 방법만을 선택한다. 지금의 선택이 앞으로 남은 선택들에 어떤 영향을 끼칠지 고려하지 않는다는 것이다.
Graph	가장먼노드, 도시분할계획, 방의개수, 순위, 커리큘럼, 팀결성(프로그래머스, 백준)	정점(V)과 간선(E)으로 이루어진 자료구조이다. 트리와는 다른점은 사이클이 존재할 수 있고 간선의 방향이 양방향일 수 있다. 종류로는 무향그래프, 유향그래프, 가중치 그래프, 사이클이 없는 방향 그래프, 완전 그래프, 부분 그래프 시간복잡도에는 정점의 갯수에 비해 간선의 개수가 적은 경우 유리한 알고리즘이다. 마찬가지로 생성자 패턴 중 빌더를 사용하여 코딩했다.
Stack/Queue	기능개발, 다리를지나는트럭, 주식가격, 프린트(프로그래머스, 백준)	Stack 은 LIFO 구조로 삽입시 stack.append(), 제거시 stack.pop()을 사용한다. Queue 는 FIFO 구조로 먼저 들어간 데이터를 생성할 시 queue.Queue(), 마지막에 들어간 데이터를 생성할시 LifoQueue, 우선순위에 따를시 PriorityQueue enqueue 를 사용한다.
Heap	더맵게, 디스크컨트롤러, 이중우선순위큐(프로그래머스, 백준)	최솟값 또는 최댓값을 빠르게 찾아내기 위해 완전이진트리 형태로 만들어진 자료구조이다.부모노드는 항상 자식노드보다 우선순위가 높다. 힙은 우선순위 큐를 구현하는데 기반이 되었다. 알고리즘은 우선순위 큐와 peek(),poll()을 사용하여 구현했다.

Sort	H 인덱스, K 번째수, 가장큰수, 두배열의원소교체, 버블정렬, 선택정렬, 성적이낮은순서, 위에서아래로, 퀵정렬, 합병정렬(프로그래머스, 백준)	두개의 배열을 비교하여 임시 저장소를 만들고 swap 한다. k 번째수는 2 차원 배열(매트릭스) 의 구조를 이해하고 2 차원 배열을 비교하여 구현했다. 합병정렬은 최소 단위로 쪼갠 뒤 다시 머지하여 구현했다.(재귀함수 호출)
Binary search	떡볶이만들기, 부품찾기, 입국심사, 징검다리(프로그래머스, 백준)	이진 탐색 알고리즘은 오름차순으로 정렬된 리스트에서 특정한 값의 위치를 찾는 알고리즘이다. 처음 중간의 값을 임의의 값으로 선택하여, 그 값과 찾고자 하는 값의 크고 작음을 비교하는 방식을 사용한다. 이때, 처음 선택한 중앙값이 만약 찾는 값보다 크면 그 값은 새로운 최댓값이 되며, 작으면 그 값은 새로운 최솟값이 된다
Dynamic	N 으로표현, 개미전사, 도둑질, 등굣길, 바닥공사, 일로만들기, 정수삼각형, 호율적인화폐구성(프로그래머스, 백준)	Top-Down 방식 - 큰 문제에서 작은 부분 문제를 재귀적으로 호출하여 리턴된 값으로 문제를 해결한다. Bottom-Up 방식 - 작은 부분 문제를 미리 계산 해두고 이 문제들을 모아 큰 문제를 해결하는 방식이다.
BFS/DFS	네트워크, 단어변환, 미로탈출, 여행경로, 음료수얼려먹기, 타겟넘버(프로그래머스, 백준)	DFS 는 그래프에서 깊은 부분을 우선적으로 탐색하는 알고리즘이다. 탐색 시작 노드를 스택에 삽입하고, 방문처리한다. 스택의 최상단 노드에 방문하지 않은 인접 노드가 있으면 그 인접 노드를 스택에 넣고 방문 처리하고, 방문하지 않은 인접노드가 없으면 스택에서 최상단 노드를 꺼낸다. 위의 과정을 더 이상 수행할 수 없을 때까지 반복한다. = BFS 는 가까운 노드부터 탐색하는 알고리즘이다. 탐색 시작 노드를 큐에 삽입하고 방문처리한다. 큐에서 노드를 꺼내 해당 노드의 인접 노드 중에서 방문하지 않은 노드를 모두 큐에 삽입하고 방문 처리한다. 위의 과정을 더 이상 수행할 수 없을 때까지 반복한다
Bruteforce	N 으로표현, 개미전사, 도둑질, 등굣길, 바닥공사, 일로만들기, 정수삼각형,	알고리즘의 가장 큰 특징은 모든 영역을 전체 탐색하는 방법이다. 전체 탐색하는 방법으로는 선형 구조를 전체적으로

	호율적인화폐구성(프로그래머스, 백준)	<p>탐색하는 순차 탐색, 비선형 구조를 전체적으로 탐색하는 깊이 우선 탐색(DFS), 너비 우선 탐색(BFS) 가 기본적인 도구이다.</p> <p>어떤 방식으로든 전체 탐색으로 문제를 해결한다면 브루트 포스 알고리즘으로 풀었다고 할 수 있다.</p> <p>주어진 문제를 선형 구조로 구조화 한다.</p> <p>구조화된 문제공간을 적절한 방법으로 해를 구성할 때까지 탐색한다.</p> <p>구성된 해를 정리한다.</p> <p>브루트포스 알고리즘의 방법으로는 for 문을 이용한 탐색, 백트래킹(재귀)을 이용한 탐색, DFS&amp;BFS 탐색 등이 있다.</p>
Dijkstra	미래도시, 전보(백준)	<p>다익스트라는 출발점에서 목표점까지의 최단거리를 구할 때 사용하는 알고리즘이다.</p> <p>distance 변수는 각각 노드까지의 최단거리를 의미한다.</p> <p>check 변수는 각각 노드를 방문했는지 표시한다.</p> <p>distance 는 처음에 나올 수 있는 가장 큰 값으로 초기화한다.</p> <p>시작노드의 거리를 0 으로 표시하고 시작노드의 check 값을 true 로 바꿔준다.</p> <p>시작노드와 연결되어 있는 노드들의 distance 값을 갱신한다.</p> <p>방문하지 않은 노드 중 distance 값이 최소인 노드를 찾는다</p> <p>distance 값이 최소인 노드의 check 값을 true 로 변경한다.</p> <p>최소 노드와 연결된 노드들(방문하지않은) distance 값을 갱신한다.</p> <p>이때 최소 노드와 연결된 distance 값이 <math>\text{distance}[\text{min\_node}] + (\text{min\_node} \text{ 와 그 노드의 거리})</math> 보다 큰 경우 distance 값을 <math>\text{distance}[\text{min\_node}] + (\text{min\_node} \text{ 와 그 노드의 거리})</math>로 갱신한다</p> <p>모든 노드를 방문할 때까지 위 방식을 반복한다</p>