그리디 알고리즘 – 최적화 문제를 해결하는 알고리즘  
(입력) 데이터 간의 관계를 고려하지 않고 수행 과정에서 욕심내어 최소값 또는 최대값을 가진 데이터를 선택  
그리디 알고리즘은 한 번 선택하면, 절대로 번복하지 않는다.  
이러한 특성으로 매우 단순하며, 제한적인 문제만이 그리디 알고리즘으로 해결된다

동전 거스름돈 문제  
최소 동전 수를 찾는 그리디 알고리즘  
동전의 단위가 정수배로 끝나야 가능하다. Ex) 500, 100, 50, 10

최소 신장 트리(Minimun Spanning Tree)  
찾는 방법 – 사이클이 없도록 모든 점을 연결시킨다.  
1) 크러스컬 알고리즘 – 가중치가 가장 작은 간선이 사이클을 만들지 않을 때에만 욕심내어 그 간선을 추가  
시간 복잡도: O(mlogm) m: 입력 그래프에 있는 간선의 수  
2) 프림 알고리즘 – 임의의 점 하나를 선택한 후, (n-1)개의 간선을 하나씩 추가시켜 트리를 만든다.  
프림 알고리즘은 트리 밖에 있는 점을 항상 연결하기 때문에 사이클이 발생하지 않는다.  
시간 복잡도 O(n2) 이지만 최소 힙을 사용하면 O(nlogn)이다.  
문제: 다음에 추가되는 점이 무엇인가

크러스컬은 n개의 트리가 한 개로 합쳐지는 것  
프림은 1개의 트리가 자라나는 것

최단 경로 문제- 주어진 가중치 그래프에서 어느 한 출발점에서 또 다른 도착점까지의 최단 경로를 찾는 문제  
가장 대표적인 알고리즘 – 다익스트라 최단 경로 알고리즘  
다익스트라 알고리즘- 출발점으로부터 최단 거리가 확정되지 않은 점들 중에서 출발점으로부터 가장 가까운 점을 추가하고, 그 점의 최단거리를 확정

문제: 그래프를 보여주고 다음 확정된 점은 무엇인지  
어떤 점으로 갈려면 어디서 가야 가장 비용이 싼지  
시간복잡도: O(n2) 최소 힙을 사용하면 O(nlogn)이다.

배낭 문제- 최대의 가치를 갖도록 배낭에 넣을 물건들을 정하는 문제  
부분 배낭 문제– 물건을 부분적으로 담는 것을 허용, 그리디로 해결  
단위 무게 당 가장 값나가는 물건 넣기  
0-1 배낭 문제– 물건을 통째로 배낭에 넣어야 함. 동적 계획, 백트래킹, 분기 한정 기법으로 해결  
시간 복잡도 O(nlogn)

집합 커버 문제 – 집합 F에서 선택하는 집합들의 수를 최소화하는 문제  
ex) 학교 배치, 어디에 배치해야 학교의 수가 최소가 되는가.  
시간복잡도 O(n3)

작업 스케줄링 문제 – 작업의 수행 시간이 중복되지 않도록 모든 작업을 가장 적은 수의 기계에 배정하는 문제  
시간복잡도 O(nlogn) + O(mn), m= 사용된 기계의 수

허프만 압축 – 각 문자를 8bit 아스키 코드로 저장, 그 파일의 비트 수는 8 \* 파일의 문자 수  
메모리 공간을 효율적으로 사용할 수 있고, 파일 전송 시간을 단축  
시간 복잡도 O(nlogn)

동적 계획(DP) 알고리즘  
입력 크기가 작은 부분 문제들을 해결한 후, 그 해들을 이용하여 더 큰 부분 문제를 해결하여 최종적으로 주어진 입력의 문제 해결  
DP 알고리즘에는 부분 문제들 사이에 함축적 순서가 존재한다.  
플루이드-와샬 알고리즘은 O(n3)의 시간복잡도를 가진다.

NP문제 안에 P문제와 NP-완전 문제가 포함되어있다. 두 문제는 다른 문제  
같은 NP-완전 문제라면 변환해서 해결할 수 있다.   
문제: A문제를 B문제로 변환해보아라.

NP-완전 문제는 NP-하드 문제이자 NP문제이다  
NP-완전 문제는 서로의 알고리즘으로 변환해서 해결할 수 있다.  
아직까지 그 누구도 이 문제들을 다항식 시간에 해결할 수 없다고 증명하지 못했다.

해결하려면 3가지중 1가지는 포기해야한다.  
1. 다항식 시간에 해를 찾는 것  
2. 모든 입력에 대해 해를 찾는 것  
3. 최적해를 찾는 것

근사 알고리즘은 NP-완전 문제를 해결하기 위해 3번을 포기한다.  
근사 알고리즘은 근사해와 근사 비율을 같이 제시해야한다.  
근사 비율이 1.0에 가까울수록 정확도가 높은 알고리즘  
근사 비율을 계산하려면 최적해를 알아야 하는 모순 발생  
최적해를 대신할 간접적인 최적해를 찾는다.

여행자 문제 – MST를 이용해서 근사해를 구한다.  
mst의 간선의 합보다 tsp의 간선의 합이 더 크다.

정점 커버문제 – 극대 매칭을 이용하여 근사해를 구한다.

통 채우기 문제  
그리디 알고리즘 방법 4가지  
- 최초 적합  
- 다음 적합  
- 최선 적합  
- 최악 적합  
문제: 특정 적합일 때 어떤 식으로 통을 채워야 하는가

클러스터링 문제 – 가장 큰 반경을 가진 그룹의 직경이 최소가 되도록 선택  
1번 센터는 랜덤으로 결정  
거리를 비교해서 가까우면 갱신  
가장 먼 점을 2번 센터로 설정  
반복?

알려진 5가지 문제는 근사비율이 2.0 이다.