

안녕하세요, 동계 대학생 S/W 알고리즘 특강의 세번째 시간인 오늘은 완전 탐색 & 그리디에 대해 다루어보도록 하겠습니다.

1. 기초 강의

동영상 강의 콘텐츠 확인 > 7. 그리디

Link :

https://swexpertacademy.com/main/learn/course/subjectDetail.do?courseId=CONTENTS_REVIEW&subjectId=AYVXpFT6QtwDFARs

※ 출석은 강의 수강 내역으로 확인합니다.

2. 실전 강의

2-1. 완전탐색

완전탐색은 문제를 해결하기 위한 간단하고 쉬운 접근법입니다. 대부분의 문제에 적용 가능하고 상대적으로 빠른 시간 안에 알고리즘을 설계하여 문제를 해결 할 수 있습니다. 문제에 포함된 자료의 크기가 작을 때 더욱 유용하고, 학술적 또는 교육적 목적에서 알고리즘의 효율성 판단을 위한 척도로 사용됩니다.

다음은 자료들의 리스트에서 키 값을 찾기 위해 완전탐색을 사용하는 예시입니다.

EX)

SequentialSearch(A[0 .. n], k)

$A[n] \leftarrow k$

$i \leftarrow 0$

while $A[i] \neq k$

$i++$

if $i < n$: return i

else : return -1

완전탐색은 모든 경우의 수를 생성하고 테스트하기 때문에 수행 속도는 느리지만 해답을 찾아내지 못할 확률이 적습니다. 이를 기반으로 하여 그리디 기법이나 동적 계획법과 조합하여 효율적인 알고리즘을 만들 수 있습니다.

2-2. 그리디

그리디 알고리즘은 최적 해를 구할 때 사용하는 근시안적인 방법입니다. 여러 경우 중 하나를 선택할 때마다 그 순간에 최적인 해를 선택해 나가면서 최종적인 해답에 도달합니다. 각 선택 시점에서 이루어지는 결정이 지역적으로는 최적이라면, 그 결정들이 모여서 만든 답이 최적 해라는 보장이 있어야 그리디 알고리즘으로 문제를 접근할 수 있습니다. 그래서, 한번 선택한 결정은 번복하지 않기 때문에 대부분의 그리디 알고리즘은 단순하며 제한적인 문제들에 적용됩니다.

그리디 알고리즘의 동작 과정은 다음과 같습니다.

- 1) 해 선택: 현재 상태에서 부분 문제의 최적 해를 구한 뒤, 이를 부분 해 집합에 추가한다.
- 2) 실행 가능성 검사: 새로운 부분 해 집합이 실행가능한지 확인한다. (문제의 제약 조건을 위반하지 않는 지 검사한다.)
- 3) 해 검사: 새로운 부분 해 집합이 문제의 해가 되는지 확인한다. 아직 전체 문제의 해가 완성되지 않았다면 1) 해 선택부터 다시 시작한다.

그리디 알고리즘은 두가지 필수 요소가 있습니다. 첫번째는 탐욕적 선택 속성(greedy choice property)입니다. 탐욕적 선택은 최적 해로 갈 수 있음을 보이는 것이고, 탐욕적 선택은 항상 안전하다는 것을 확인할 수 있어야합니다. 두번째는 최적 부분 구조(optimal substructure property)입니다. 최적화 문제를 정형화 해야한다는 의미이고, 하나의 선택을 하면 풀어야 할 하나의 하위 문제가 남아야 합니다. 정리하면 '원 문제의 최적 해 = 탐욕적 선택 + 하위 문제의 최적 해'임을 증명해야 합니다.

3. 기본 문제

- 최장 공통 부분 수열
- 0/1 Knapsack