

시뮬레이션

시뮬레이션이란?

- 시뮬레이션은 문제에서 주어진 상황을 이해하고 구현하는 과정이다. 그렇기에 문제를 풀 때는 문제를 구조화하고 기법보다 구현 방식을 생각할 필요가 있다.
- 시뮬레이션 문제의 접근 방식
 - 하나의 문제를 여러 개의 부분 문제로 분리한다.
 - 예외 처리가 필요하면 독립 함수로 구현하기. 구현할 부분이 많기에 미리미리 분리시켜 놓지 않으면 헛갈린다.
- 시뮬레이션 문제는 선형대수와 연관이 깊다. 특히, 행렬 연산을 많이 다루니까 기초적인 행렬 연산을 수행할 수 있어야 한다.

행렬 연산

- 덧셈, 뺄셈, 곱셈, 전치 행렬 등. 기본적인 행렬 연산을 수행하는 문제가 나온다.

좌표 연산

- 조건에 따라 물체가 이동하는 과정이 있는 문제인 경우, 좌표 연산을 사용한다.
- 좌표 연산은 일단 각 좌표를 배열로 표현하고 이동 방식을 함수나 배열로 정의해서 이를 적용해 나간다.

대칭, 회전 연산

- 좌우 또는 상하 대칭의 경우 축이 행이나 열이느냐에 따라 해당하는 축의 대칭값으로 계산하면 된다.
- 회전 연산의 경우에는 배열을 90도 돌렸을 때의 연산값을 기준으로 삼아서 회전을 수행하면 된다.

풀어보자

- 이진 변환, <https://school.programmers.co.kr/learn/courses/30/lessons/70129>
- 카펫, <https://school.programmers.co.kr/learn/courses/30/lessons/42842>
- 캐릭터의 좌표, <https://school.programmers.co.kr/learn/courses/30/lessons/120861>

그리디

그리디란?

- 해를 구할 때마다 현재 주어진 조건 하에서 최적의 해를 고르는 것. 경우의 수를 전부 따지면서 그 중에 최적해를 구하는 게 아닌 그 때 그 때마다 눈에 보이는 최적해를 구한다.

그리디 알고리즘 사용 조건

- 최적 부분 구조(optional substructure) : 부분해를 푸는 과정이 최적해를 구하는 과정과 일치
- 그리디 선택 속성(greedy selection property) : 선택 과정이 다른 과정에 영향을 주지 않음

그리디 적용 : 최소 신장 트리(MST)

- 신장 트리(spanning tree)는 모든 정점이 간선으로 연결되어 있고 간선 개수가 정점 개수보다 하나 적은 그래프
- 최소 신장 트리(Minimum Spanning Tree) : 신장 트리에서 간선의 가중치 합이 최소인 트리. 항공기의 운항 경로 최적화나 네트워크 분야에서 많이 활용한다.
- 프림 알고리즘(Prim's Algorithm)
 - 임의의 정점을 하나 선택해서 최소 신장 트리에 추가.
 - (그리디)최소 신장 트리와 연결되어 있는 정점 중 가중치가 가장 작은 정점을 최소 신장 트리에 추가. 단, 순환 구조를 낳으면 안 됨.
 - 그리디 단계를 신장 트리 조건 만족 시까지 반복한다.
 - 시간 복잡도 : $O(E \cdot \log V)$, 인접 행렬 활용 시 $O(N^2)$
- 크루스칼 알고리즘
 - 그래프의 모든 간선을 가중치 기준으로 오름차순 정렬
 - (그리디)가중치가 낮은 간선부터 최소 신장 트리에 하나씩 추가(그리디). 단, 순환 구조를 낳으면 안 됨.
 - 그리디 단계를 신장 트리 조건을 만족할 때까지 반복한다.
 - 시간 복잡도 : $O(E \cdot \log E)$

그리디 적용 : 배낭 문제

- 배낭에 담을 수 있는 최대 무게가 존재하며 무게와 가치가 다른 짐이 있음. 이 때, 배낭의 무게를 초과하지 않으면서 배낭에 넣은 짐들의 가치를 최대로 하는 문제.
- 부분 배낭 문제(fractional knapsack problem) : 짐을 쪼갤 수 있는 경우
 - 짐별로 무게당 가치를 구한다.
 - (그리디)무게당 가치가 높은 짐부터 넣을 수 있는 만큼 배낭에 넣는다.
 - 배낭 용량이 짐 무게보다 크면 짐을 쪼개서 넣는다.
 - 그리디 단계를 배낭이 허용하는 용량이 0이 될 때까지 수행한다.
 - 시간 복잡도 : $O(N \cdot \log N)$
- 0/1 배낭 문제(0/1 knapsack problem) : 짐을 쪼갤 수 없는 경우
 - 쪼갤 수 없으므로 무조건 무게당 가치가 높은 짐들이 배낭에 채워넣었을 때 가치가 높은 값이라 예측할 수 없다. 그렇기에 그리디가 아닌 동적 계획법(DP)으로 풀어야 한다.

Quiz

- 쿨 고르기, <https://school.programmers.co.kr/learn/courses/30/lessons/138476>

- 기지국 설치, <https://school.programmers.co.kr/learn/courses/30/lessons/12979>