P10. Shiritori

21500800 홍순규

문제해석

주어진 단어들을 모두 사용해 끝말잇기가 가능한지 판정하고, 사전 순으로 가장 잘 정렬된 답을 출력한다

아이디어

- 단어의 마지막 글자가 다음 단어의 첫 글자와 같으면 잇는 Directed Graph로 문제를 표현한다
- 단어를 먼저 정렬하고 정렬된 순서로 탐색하여, 가장 먼저 발견된 답이 정답이 될 수 있게 한다
- 계산 시간을 줄일 수 경우들을 생각한다

기본 구현

문제를 표현한 Directed Graph를 깊이 우선 탐색하고, node를 방문할 때마다 path set에 기록한다
Edge가 없는 node에 이르렀을 때 Path가 전체 그래프가 아닌 경우 back tracking 한다
시작점은 정렬된 순으로 고르고, 한 node와 연결된 node들을 고르는 순서도 정렬된 순으로 정한다

계산 시간을 줄이는 방법

1. 끝말잇기는 첫 번째 단어의 첫 글자와, 마지막 단어의 마지막 글자를 제외하고, 모든 단어들의 첫 번째 글자는 다른 단어들의 마지막 글자와 일 대 일 매칭 관계일 때 성립한다

2. 첫 번째 단어의 첫 글자와 마지막 단어의 마지막 글자가 다르다면, 첫 번째 단어와 마지막 단어를 고정할 수 있다

3. 한 node에서 파생할 수 있는 모든 path를 이미 탐색한 경우, 그 node에서 출발하여 방문할 수 있는 최대 node의 개수 M을 알고 있다. 다음 번에 방문할 node가 탐색했던 노드라면, 지금까지의 path set의 size + 1 + M이 전체 단어의 수 N 보다 작을 때 방문할 필요가 없어진다 (전체 단어를 사용한 끝말잇기가 불가능하다)

자료구조

1. 단어의 연결관계를 표현한 인접리스트
2. DFS의 진행과정을 나타내는 Path set
3. 각 Node에서 도달 가능한 최대 node의 개수를 저장한 일차원 배열

시간 복잡도

기본 구현에 따라, 전체 node 각 각에 대해 DFS를 돌리기 때문에 시간복잡도는 O(N * (V+E))이 된다

또한, 한 node에 방문할 때 해당 node가 이미 path에 있는지 확인하는 것을 최적화하지 못했기 때문에 N이 곱해져 O(N^2 * (V+E))의 시간 복잡도를 가진다

계산 시간을 줄이는 3번째 방법을 적용하면 각 node 당 한 번씩만 DFS로 방문하므로, 다시 O(N * (V+E))로 줄일 수 있다.