

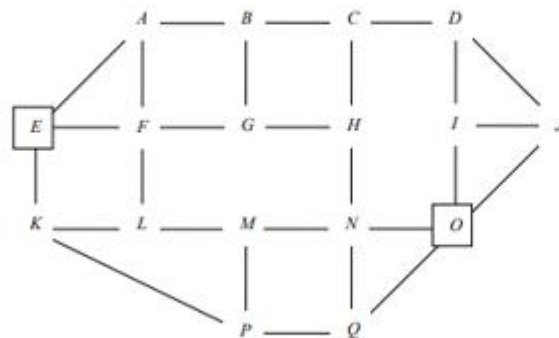
Homework #1

인공지능

정보컴퓨터공학부

201924548 이풍헌

1. 아래 그래프의 node E로부터 node O까지 가는 path를 찾고자 한다. 아래 물음에 답하라. 단, step cost는 모두 동일하고, node를 expand할 때 생성되는 child의 순서는 node 이름의 alphabet 순을 따르는 것으로 가정한다.



- (a) Breadth-first search (BFS) 알고리즘을 구현한다면 TREE-SEARCH 보다 GRAPH-SEARCH를 기반으로 하는 것이 더 좋은 이유는 무엇인가? 시간과 공간 양 측면에서 모두 설명하라.

tree search에서는 explored set을 저장하지 않기에 이미 방문한 노드에 대해서도 새로운 path를 만들어 탐색한다. 반면 graph search는 explored set을 저장해 방문한 노드에 대해서는 새로운 path를 만들지 않으므로 탐색하는 노드의 수가 줄어들게 되므로 이는 시간, 공간적 면에서 더 효율적이라 할 수 있다.

- (b) CH01 강의자료 28 Page의 BFS 알고리즘은 GRAPH-SEARCH를 기반으로 하고 있지만 20 Page의 GRAPH-SEARCH와는 다른 점이 있다. 어떤 점이 다르며 그 이유는 무엇인가?

28page의 bfs는 child node를 frontier에 추가할 때 goal state를 테스트해 불필요한 depth탐색을 줄여 메모리,시간적으로 절약할 수 있다.

- (c) CH01 강의자료 28 Page의 BFS 알고리즘을 적용하라. 매 iteration마다 frontier로부터 노드 하나를 꺼내기 직전 시점에서의 frontier와 explored set 내용을 확인하여 아래 표를 완성하고, 최종적으로 찾은 solution(path)을 제시하라. 이 표에서 frontier 내의 각 node는 search path가 어떻게 형성되어 가고 있는지 알 수 있도록 partial path의 형태로 되어 있음에 유의하라. 처음 3 iteration에 대한 답은 예시로 주었음.

Frontier	Explored set
E	

AE,FE,KE	E
FE,KE,BAE	E,A
KE,BAE,GFE,LFE	E,A,F
BAE,GFE,LFE,PKE	E,A,F,K
GFE,LFE,PKE,CBAE	E,A,F,K,B
LFE,PKE,CBAE,HGFE	E,A,F,K,B,G
PKE,CBAE,HGFE,MLFE	E,A,F,K,B,G,L
CBAE,HGFE,MLFE,QPKE	E,A,F,K,B,G,L,P
HGFE,MLFE,QPKE,DCBAE	E,A,F,K,B,G,L,P,C
MLFE,QPKE,DCBAE,NHGFE	E,A,F,K,B,G,L,P,C,H
QPKE,DCBAE,NHGFE	E,A,F,K,B,G,L,P,C,H,M

QPKE에서 child node가 OQPKE 노드를 생성하고 frontier에 추가 전 goal state 체크에서 찾기에 탐색이 종료됨

(d) TREE-SEARCH를 기반으로 depth-first search (DFS) 알고리즘을 구현할 경우 GRAPH-SEARCH를 기반으로 하는 것에 비해 어떤 장점이 있는가? 또한 주의해야 할 점은 무엇인가?

Dfs를 tree search로 구현하면 중복방문을 허용하지 않기에 loopy path에 빠지지 않는다. 또한 방문한 node를 기록할 필요가 없기에 메모리 사용량도 적다. 하지만 Tree의 depth가 유한한지 체크하지 않으면 비효율적인 방법이 될 수 있다.

(e) 변형된 TREE-SEARCH 기반의 DFS가 진행되어 감에 따라 *frontier*가 어떻게 변해 가는지를 위의 (c)에서처럼 partial path의 형태로 표시하고 최종적으로 찾은 solution을 제시하라.

FRONTIER
E
AE
BAE
CBAE
DBAE
IDBAE
JIDBAE
OJIDBAE

Solution path는 OJIDBAE가 된다.

(f) Iterative deepening search 를 이용하여 node E로부터 node M까지의 (주의: O까지가 아님) path 를 찾을 경우 생성되는 search tree 를 각 depth limit 별로 보이고 (단계별로 모두 그리지 않아도 됨) 탐색 중 생성된 node 의 총 수를 세어 보라. (단, root node 는 총 수에 포함이 안된다.)

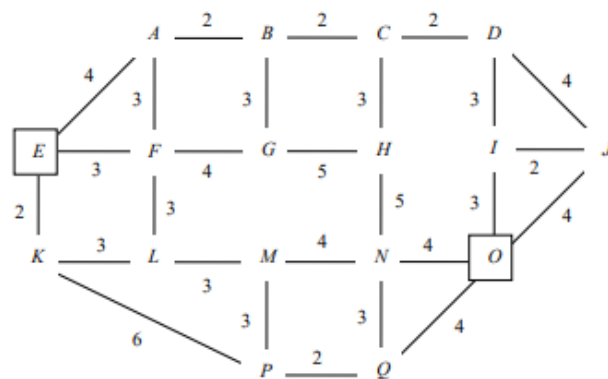
Depth limit 1일 경우 3개의 노드

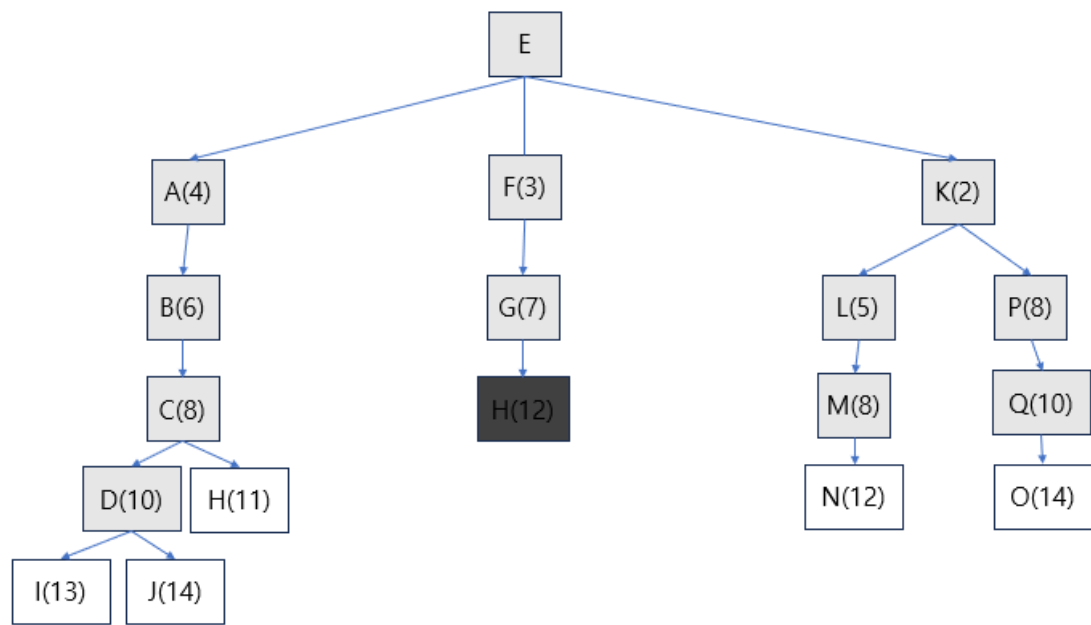
Depth limit 2일 경우 3+7개의 노드

Depth limit 3일 경우 3+7+9의 노드가 생성된다.



2. 다음 그래프에서 node E로부터 node O까지 가는 최소 비용의 path를 찾기 위해 변형된 GRAPH-SEARCH 기반의 uniform-cost search를 수행할 경우 생성되는 search tree를 보이고 (단계별로 그리지 않아도 됨) 탐색 중 생성된 node의 총 수를 세어 보라. 단, step cost는 그래프의 각 edge에 표시된 바와 같고, 각 node를 expand할 때 생성되는 child의 순서는 node 이름의 alphabet 순을 따르며, 동일한 평가 값을 가지는 node들을 expand하는 순서 역시 alphabet 순인 것으로 가정한다. (단, root node는 총 수에 포함이 안된다.)





Uniform cost search로 실행할 경우의 tree이다. 총 생성 노드는 17개이다.