Lecture Title: Depth-First Search Algorithm.

*Depth First Search Algorithm*

Initialize: open=[start]; closed=[ ]; parent[start]=null; found=no;
While open<>[ ] do
  Begin
    - Remove the first state from left of open, call it X;
    - If X is a goal then {found=yes; break}
    - Generate all possible children of x and put them in list L;
    - Put X on closed;
    - Eliminate from L any child already on closed;
    - Eliminate from open any child in L;
    - For each child y in L set parent[y]=X;
    - Add L to the left of open;
  End.
If found =yes then
  Generate and return the solution path.
Else
  Output no solution.
Example: use depth first search to find the path between A and L for the following search space:

Solution:
Start=[A], goal=L.

Iteration #0:
Open=[A], closed=[ ], parent[A]=null.

Iteration #1:
X=A, L=[B,C,D,E], open=[B,C,D,E], closed=[A].

Iteration #2:
X=B, L=[F,C], open=[F,C,D,E], closed=[A,B].
Parent[F]=B, parent[C]=B.

Iteration #3:
X=F, L=[I,J], open=[I,J,C,D,E], closed=[A,B,F].

Iteration #4:
X=I, L=[N], open=[N,J,C,D,E], closed=[A,B,F,I].
Parent[N]=I.

Iteration #5:
X=N, L=[ ], open=[J,C,D,E], closed=[A,B,F,I,N].

Iteration #6:
X=J, L=[ ], open=[C,D,E], closed=[A,B,F,I,N,J].

Iteration #7:

Iteration #8:

Iteration #9:

Iteration #10:
X=L
Since L is a goal, stop and find path.

Path: A→B→C→G→L