

Lecture 5 (Uninformed Search cotd.)

1 Defining Node

```
function childNode(problem, parent, action):  
    return node (  
        STATE = problem.result(parent.STATE, action)  
        PARENT = parent  
        ACTION = action  
        COST = parent.COST + problem.STEP_COST(parent.STATE, action)  
    )
```

2 Graph Search (modification to Tree Search)

```
function GraphSearch(problem, strategy):  
    initialise frontier with initial state  
    initialise explored set to EMPTY  
    while true:  
        if frontier is EMPTY, return failure  
        choose a leaf node and remove it from frontier  
        if node contains goal state return solution  
        add node to explored set  
        expand node and add resulting nodes to frontier  
            only if not in frontier or explored set
```

3 Measuring Performance

1. b is branching factor
2. m is maximum depth
3. d is depth of shallowest goal

4 Properties of Search Algorithms

1. Completeness

2. Optimality
3. Time complexity
4. Space complexity

5 Types of Search Algorithms

1. Breadth First Search:
 - i. Expand the shallowest unexplored node
 - ii. The frontier is a queue
 - iii. Analysis -
 - Complete
 - Optimal (finds the closest solution)
 - Time complexity = $O(b^d)$
 - Space complexity = $O(b^d)$, very huge
2. Depth First Search:
 - i. Expand the deepest unexplored node
 - ii. Analysis
 - Complete
 - Not optimal (finds the leftmost solution)
 - Time complexity = $O(b^m)$
 - Space complexity = $O(b \cdot m)$