**UCS vs Iterative Lengthening Search**

**UCS**: **Uniform-cost search** is an uninformed search algorithm that uses the lowest cumulative cost to find a path from the source to the destination. Nodes are expanded, starting from the root, according to the minimum cumulative cost. The uniform-cost search is then implemented using a Priority Queue.

**UCS Algo**: 1. Insert the root node into the priority queue

2. Repeat while the queue is not empty:  
Remove the element with the highest priority.  
If the removed node is the destination, print total cost and stop the algorithm.  
Else, enqueue all the children of the current node to the priority queue, with their cumulative cost from the root as priority.

**Iterative Lengthening Search**: **Iterative lengthening search** an iterative analogue to uniform cost search. The idea is to use increasing limits on path cost. If a node is generated whose path cost exceeds the current limit, it is immediately discarded. For each new iteration, the limit is set to the lowest path cost of any node discarded in the previous iteration.

**ILS Algo**: 0. Set limit to zero.

1. Perform UCS on initial node while discarding any generated node with path cost greater than limit.

2. If UCS ended with Failure or Success, return that as a result.

3. Otherwise set limit to minimum path cost of generated nodes that were discarded in previous run of UCS.

4. Go to line 1.

**🡪 Observation:** In UCS each path for adjacent nodes are stored in a priority queue and then fetch the min value from the queue and continue but in ILS we set limit value which works as max path cost for a particular iteration. Limit changes as per the last shortest path find on the previous iteration. ILS iterate UCS with different values of limit and find the path.