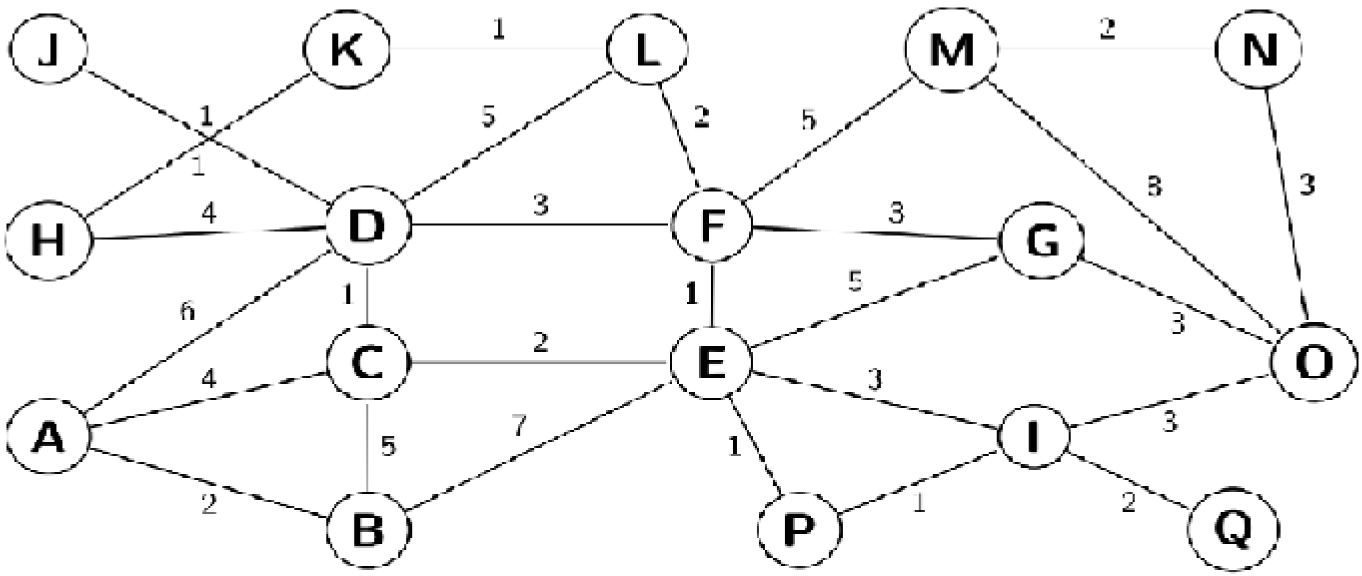
Introduction to AI - Exercise I

Tom Goodman

1 Graph



2 Question One

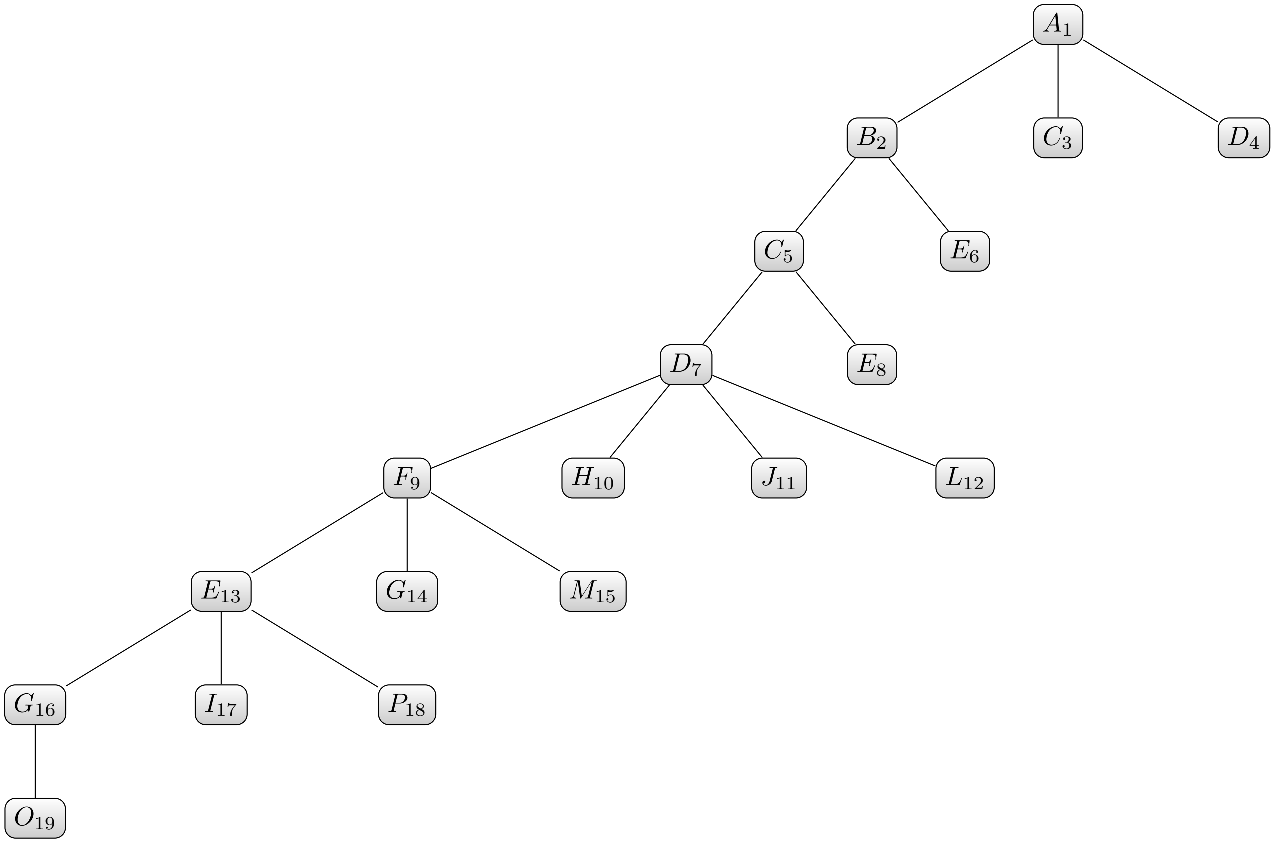
2.1 Brief

For this exercise ignore the path costs. Perform Depth-First-Search to find a path from A to O. Assume that nodes are expanded in alphabetic order. Write down carefully the values in your data structures Explored and Frontier as well as the search tree. [$30\% - 30$ *marks*]

2.2 Answer - Data Structures

Frontier	**Explored**
$[A_1]$	$[\,]$
$[B_2, C_3, D_4]$	$[A_1]$
$[C_5, E_6, C_3, D_4]$	$[A_1, B_2]$
$[D_7, E_8, E_6, C_3, D_4]$	$[A_1, B_2, C_5]$
$[F_9, H_{10}, J_{11}, L_{12}, E_8, E_6, C_3, D_4]$	$[A_1, B_2, C_5, D_7]$
$[E_{13}, G_{14}, M_{15}, H_{10}, J_{11}, L_{12}, E_8, E_6, C_3, D_4]$	$[A_1, B_2, C_5, D_7, F_9]$
$[G_{16}, I_{17}, P_{18}, G_{14}, M_{15}, H_{10}, J_{11}, L_{12}, E_8, E_6, C_3, D_4]$	$[A_1, B_2, C_5, D_7, F_9, E_{13}]$
$[O_{18}, I_{17}, P_{18}, G_{14}, M_{15}, H_{10}, J_{11}, L_{12}, E_8, E_6, C_3, D_4]$	$[A_1, B_2, C_5, D_7, F_9, E_{13}, G_{16}]$

2.3 Answer - Tree



2.4 Answer - Sequence

The following sequence is the solution to reach the goal state, $\{O\}$.

$$[A_1, B_2, C_5, D_7, F_9, E_{13}, G_{16}, O_{19}]$$

3 Question Two

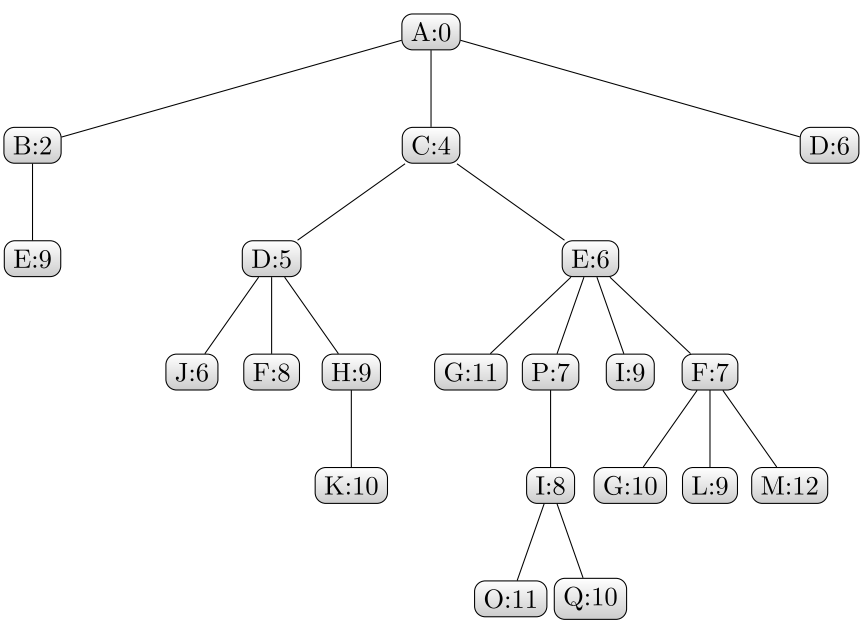
3.1 Brief

Perform Uniform Cost Search to find a path from A to O. Again assume that nodes are expanded in alphabetic order. Write down carefully the values in your data structures Explored and Frontier as well as the search tree [30% - 30 marks]

3.2 Answer -Data Structures

Frontier	**Explored**
[A:0]	[ ]
[B:2, C:4, D:6]	[A:0]
[C:4, D:6, E:9]	[A:0, B:2]
[D:5, E:6]	[A:0, B:2, C:4]
[E:6, J:6, F:8, H:9]	[A:0, B:2, C:4, D:5]
[J:6, F:7, P:7, H:9, I:9, G:11]	[A:0 B:2, C:4, D:5, E:6]
[F:7, P:7, H:9, I:9, G:11]	[A:0, B;2, C:4, D:5, E:6, J:6]
[P:7, H:9, I:9, L:9, G:10, M:12]	[A:0, B;2, C:4, D:5, E:6, J:6, F:7]
[I:8, H:9, L:9, G:10, M:12]	[A:0, B;2, C:4, D:5, E:6, J:6, F:7, P:7]
[H:9, L:9, G:10, Q:10, O:11, M:12]	[A:0, B;2, C:4, D:5, E:6, J:6, F:7, P:7, I:8]
[L:9, G:10, K:10, Q:10, O:11, M:12]	[A:0, B;2, C:4, D:5, E:6, J:6, F:7, P:7, I:8, H:9]
[G:10, K:10, Q:10, O:11, M:12]	[A:0, B;2, C:4, D:5, E:6, J:6, F:7, P:7, I:8, H:9, L:9]
[K:10, Q:10, O:11, M:12]	[A:0, B;2, C:4, D:5, E:6, J:6, F:7, P:7, I:8, H:9, L:9, G:10]
[Q:10, O:11, M:12]	[A:0, B;2, C:4, D:5, E:6, J:6, F:7, P:7, I:8, H:9, L:9, G:10, K:10]
[O:11, M:12]	[A:0, B;2, C:4, D:5, E:6, J:6, F:7, P:7, I:8, H:9, L:9, G:10, K:10, Q:10]
[M:12]	[A:0, B;2, C:4, D:5, E:6, J:6, F:7, P:7, I:8, H:9, L:9, G:10, K:10, Q:10, O:11]

3.3 Answer - Tree



3.4 Answer - Solution

The following sequence is the solution to reach the goal state, $\{O\}$.

$$ACEPIO \ (Total \ Cost \ = \ 11)$$

4 Question 3

4.1 Brief

Ignore again the path costs and perform bi-directional search to solve the problem. [30% - 30 marks]

4.2 Reasoning for the Combination of Searches

4.2.1 Breadth-First in both Directions

Although this method is guaranteed to meet at some point, which will ensure that the search is always complete, the cost of performing the search (memory) is the highest.

4.2.2 Breadth-First in one direction, Depth-First in the other

This method is guaranteed to meet, which guarantees completeness. The issue; however, is still the memory usage of a Breadth-First search. As well as this, there is the problem of a difference in memory usage depending on whether the Breadth-First search starts from the start node or the goal node, since the graph is unlikely to be symmetrical.

4.2.3 Depth-First in both Directions

This method isn't guaranteed to meet, since the search frontiers of both would be small. Were they to meet;however, the search would almost certainly have a very low time complexity. [1]

Based on the above information, and the knowledge that the search space is finite, I decided to implement a Depth-first search running in both directions.

4.3 Answer - Data Structures

Throughout this search, the algorithm checks for a 'meeting point' on generation, not expansion.

Frontier A	**Explored A**
[A]	[ ]
[B, C, D]	[A]
[C, D, E]	[A, B]

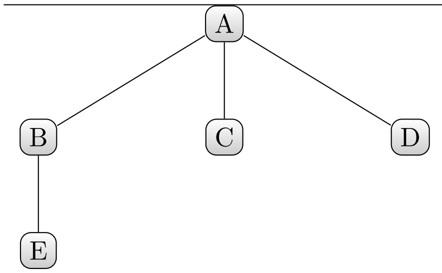
Frontier Ω	**Explored Ω**
[O]	[ ]
[G, I, M, N]	[O]
[E, F, I, M, N]	[O, G]

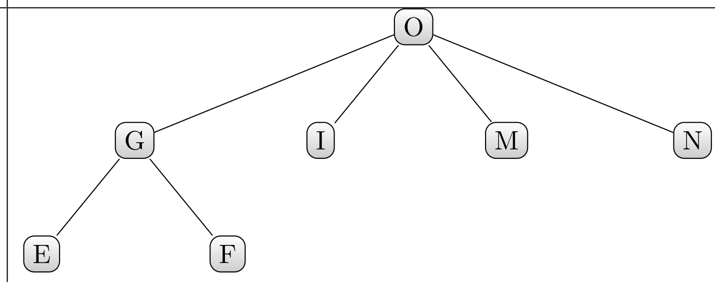
At this point, node E has been generated by both searches. This means that a path to the goal state has been found.

[1] It is worth noting that if the search space was not known to be finite, it would be logical to implement Iterative deepening rather than Depth-First search.

4.4 Answer - Trees

Tree A	**Tree** Ω





4.5 Answer - Solution

The following sequence is the solution to reach the goal state, $\{O\}$.

$$ABEGO$$

5 Question 4

5.1 Brief

Design an algorithm for bi-directional search that would allow you to search for an optimal solution with respect to path costs. [10% - 10 marks]

5.2 Answer

$f \leftarrow cumulative\ path\ cost\ from\ the\ start\ state$
$g \leftarrow cumulative\ path\ cost\ from\ the\ goal\ state$

$frontierA \leftarrow prority\ queue\ ordered\ by\ f$
$exploredA \leftarrow queue$
$frontier\Omega \leftarrow priority\ queue\ ordered\ by\ g$
$explored\Omega \leftarrow stack$

loop do
    //Checks whether a solution has been found
    for node in $frontierA$
        if node in $frontier\Omega$
            return solution
    //Expands node A
    $nodeA \leftarrow\ Pop(frontierA)$
    for childNode attached to nodeA
        Queue(childNode, $frontierA$)
    Queue(nodeA, $exploredA$)
    //Expands node Ω
    $node\Omega \leftarrow\ Pop(frontier\Omega)$
    for childNode attached to nodeΩ
        Queue(childNode, $frontier\Omega$)
    Queue(nodeA, $explored\Omega$)