

# Homework 1

Anıl Eren Göçer  
e2448397@ceng.metu.edu.tr

March 23, 2024

## Question 1

a)

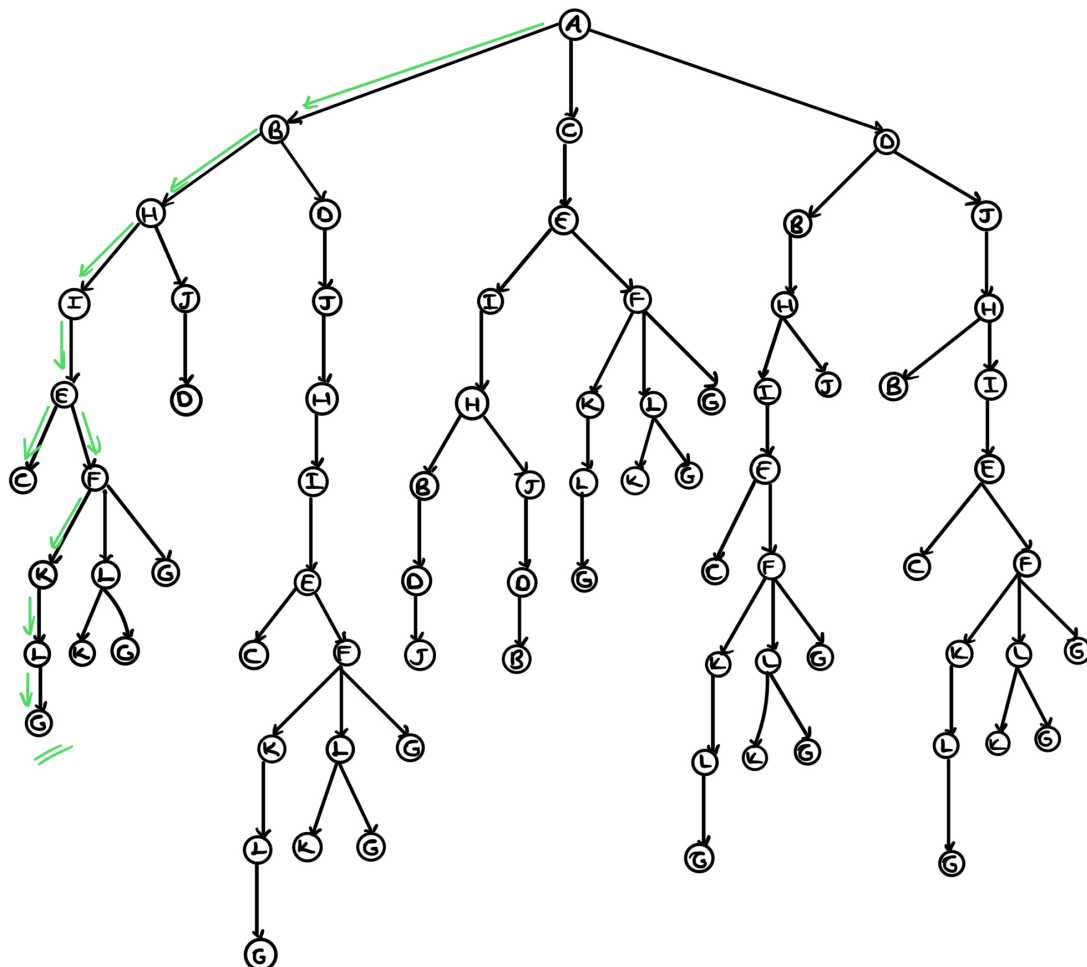
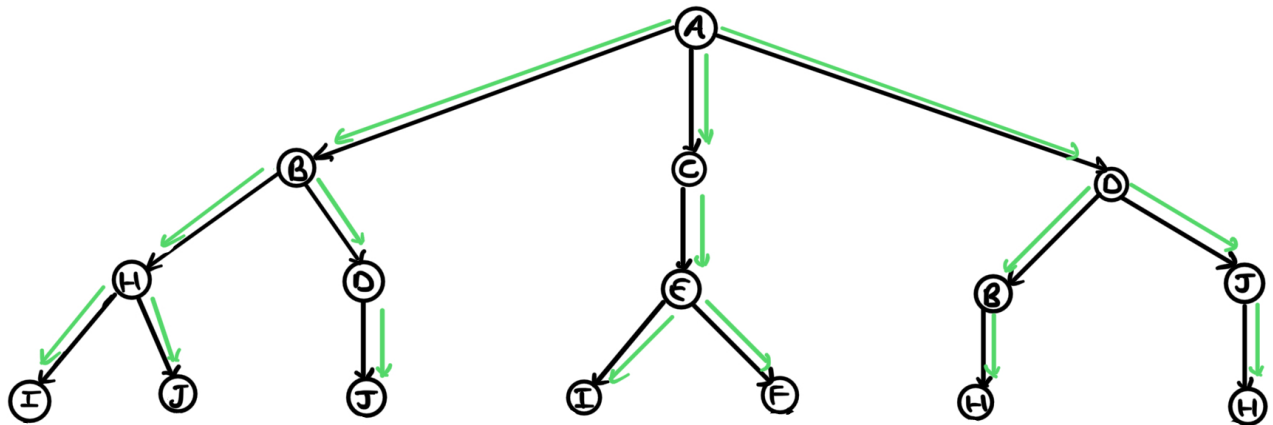


Figure 1: Depth First Search Tree

**The order of nodes visited:**  $A, B, H, I, E, C, F, K, L, G$

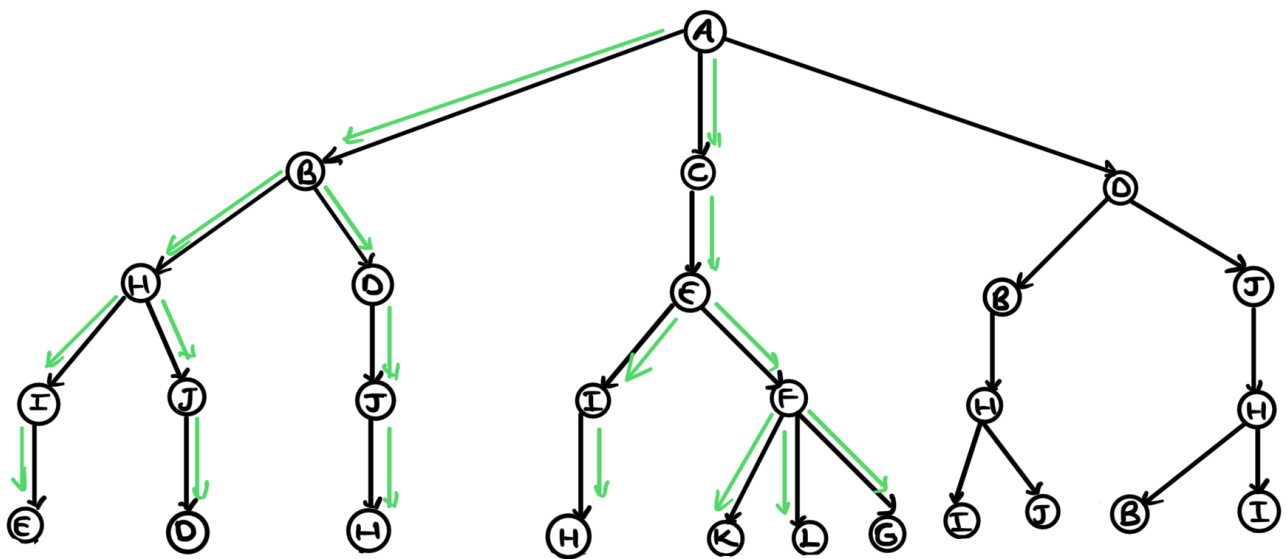
1st iteration, limit = 3



**The order of nodes visited:**  $A, B, H, I, J, D, J, C, E, I, F, D, B, H, J, H$

We didn't reach to the goal node (G), so we continue to apply iterative deepening.

2nd iteration, limit = 4



**The order of nodes visited:**  $A, B, H, I, E, J, D, D, J, H, C, E, I, H, F, K, L, G$

As you can see, we reached to the goal node (G) and stopped. Therefore, 2nd iteration , limit = 4 is the last iteration and Figure 3 shows its search tree.

## Question 2

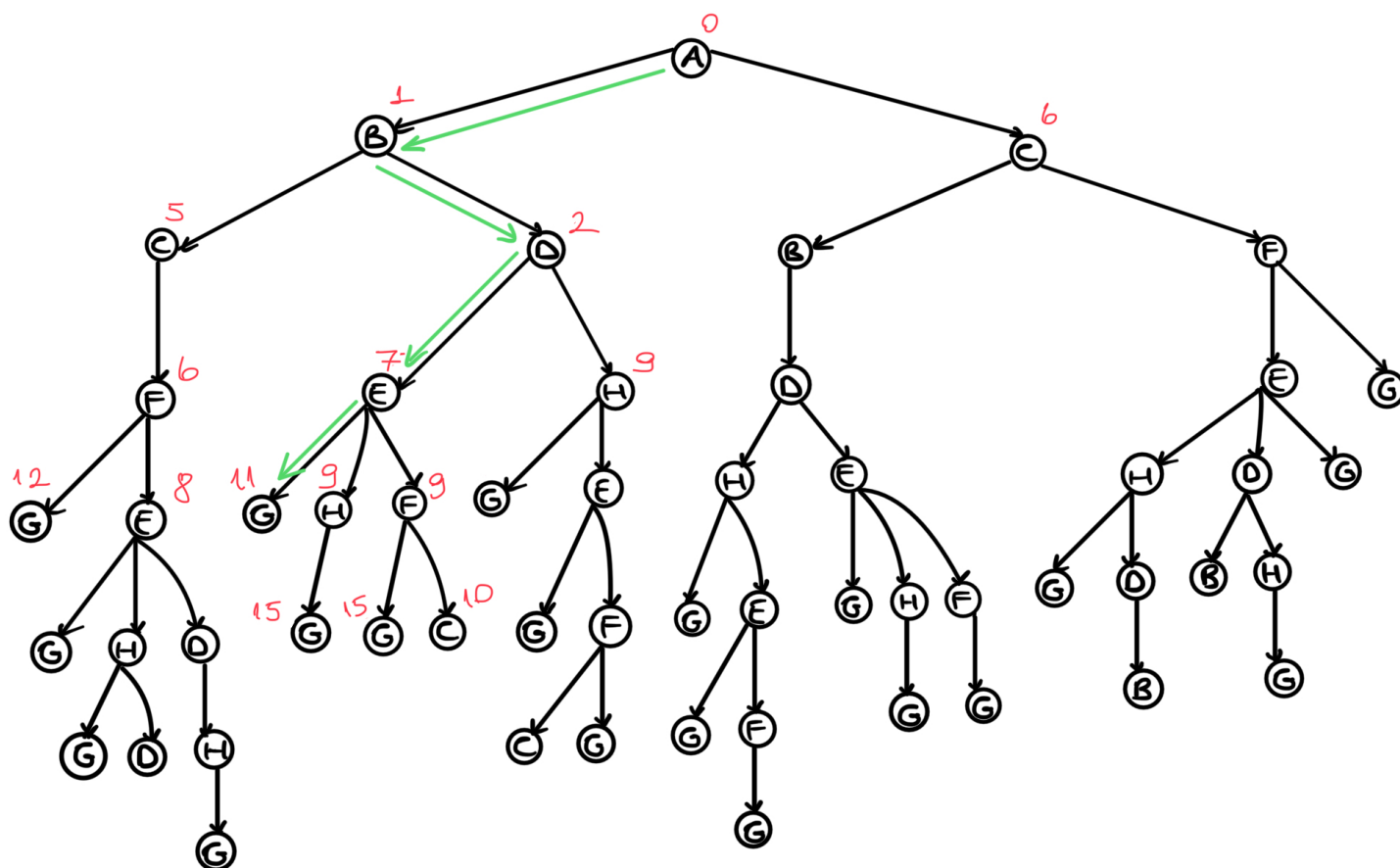


Figure 4: Search Tree Uniform Cost Search

In order to avoid confusions, I would like to give the high level overview of the algorithm that I applied. Actually, it is the one that we saw in the lectures.

---

**Algorithm 1** Uniform Cost Search

```

fringe.push((initial_node, 0))
while fringe.isEmpty() do
    current_node, cumulative_cost_of_current_node = fringe.pop()
    for child in current_node.children() do
        child_cumulative_cost = cumulative_cost_of_current_node + cost(current_node, child)
        fringe.push((child, child_cumulative_cost))
    end for
end while

```

---

**Algorithm 2** Fringe Push

---

**Parameter:** (key, value)

shouldWeAdd = True

**for** item **in** fringe.list **do**

**if** item.key == key **and** item.value < value **then**

        shouldWeAdd = False

**break**

**end if**

**end for**

**if** shouldWeAdd == True **then**

    fringe.list.overwrite((key, value))

**end if**

---

## Fringe

**Step 0:** ( $A, 0$ )

**Step 1:** ( $B, 1$ ), ( $C, 6$ )

**Step 2:** ( $D, 2$ ), ( $C, 5$ )

**Step 3:** ( $C, 5$ ), ( $E, 7$ ), ( $H, 9$ )

**Step 4:** ( $F, 6$ ), ( $E, 7$ ), ( $H, 9$ )

**Step 5:** ( $E, 7$ ), ( $H, 9$ ), ( $G, 12$ )

**Step 6:** ( $H, 9$ ), ( $F, 9$ ), ( $G, 11$ )

**Step 7:** ( $F, 9$ ), ( $G, 11$ )

**Step 8:** ( $C, 10$ ), ( $G, 11$ )

**Step 8:** ( $G, 11$ )

**Step 10:** Empty fringe

As it is described in the search tree with green arrows, **the resulting path is:**  $A, B, D, E, G$  .

**The order of nodes visited** is  $A, B, D, C, F, E, H, F, C, G$  .

Note that, the order of nodes visited is nothing but the order of popped items from the fringe at each step.