Jonathan Fokkan
260447938
COMP250
Assignment 3
Question 1 (c) & 2

1.

(c)

The queue would not be fair in terms of balancing waiting times for planes in the various queues.
A plane waiting to take off could wait longer than a plane attempting to land. Since there is nothing
special about a plane landing compared to one taking off in this example, they should try to wait for approximately
equal lengths. A better data structure for such a situation would be priority queues where priority is based upon time
of request to land or take off and some other environmental factor if need be.

2.

(a)

Considering a tree of depth d with a branching factor of greater than or equal 2.
The minimum number of nodes for such a tree is: $2^{(d+1)} - 1$
The maximum number of nodes for such a tree is: infinite.

Proof for the minimum:

Let N be the number of nodes in the tree.
Base Case:
depth 0:

```
        0
```

$N = 2^{(1)} - 1 = 1$

Depth 1:

```
        0
       / \
      0   0
```

$N = 2^{(2)} - 1 = 3$

induction Step:
Assume that $2^{(d+1)}-1$ is true for trees up to d

for a tree of depth d+1
$N = N_{left} + N_{right}$
N_{left} and N_{right} are of depth d

$N = 2^{(d+1)}-1 + 2^{(d+1)} -1 = 2^{(d + 2)} - 2$

Proof for the maximum:
Base Case:
Depth 1:
N = infinity

Induction Step:
Assume that the max of such a tree at depth d is infitite.

for a tree at depth d+1:
the Max of this tree would be the sum of max of its subtrees plus the root.
The max of it's roots is infinite by the induction hypothesis.

There the max of this tree is infinite.

(b)

Prove that there are 2^k paths from root to leaves, for a full binary tree of depth k.

Base Case:
depth 1:

```
3 nodes      0
            / \
           0   0
```

Let P = paths
P = P_left + P_right
P_left = 1
P_right = 1
2^1 = P_left+P_right =2 (correct)

Depth 2:
7 nodes

```
             0
            / \
           0   0
          / \ / \
         0  0 0  0
```

P_left = P1 = P1_left + P1_right
P1_left = 1
P1_right = 1
P_left = 2
P_right = P2= P2_left + P2_right
.
. similarly
.
P_right = 2

2^2 = P = 4 (correct)

Induction step:
Assume there are 2^k paths for binary trees of depth up to k.

```
    0
   / \
  ▲   ▲
```

for depth k+1
P = P_left + P_right
P_left = 2^(k + 1 -1)   by induction hypothesis
P_right = 2^(k + 1 -1)   by induction hypothesis
P = 2^(k) + 2^(k)
  = 2^(k + 1)

(c)

Prove that a binary tree in which each node has either 0 or 2 successors has an odd number of nodes.

Base Case:

```
    0
```

1 node
Odd number of nodes.

```
        0
       / \
      0   0
      3 nodes
      Odd number of nodes.

    Induction step:
    let N be the number of nodes in a tree.
    Assume:
    N = N_left + N_right + 1  is odd
        holds for up to N.

    N+2 is odd. N can only increase by a multiple of 2.
    The sum of odd and even numbers is odd.
```

3.

Original:

```
             15
            /  \
           7   18
          / \    \
         3  12   23
            /
           10
```

Insert(21):

```
             15
            /  \
           7   18
          / \    \
         3  12   23
            /    /
           10   21
```

Insert(16):

```
             15
            /  \
           7   18
          / \  / \
         3  12 16 23
            /    /
           10   21
```

Insert(2):

```
             15
            /  \
           7   18
          / \  / \
         3  12 16 23
        /   /    /
       2   10   21
```

remove(18):

```
        15
       /  \
      7    16
     / \     \
    3   12    23
   /   /     /
  2   10    21
```