Problem 1

To find the minimum key, choose the first child (i.e. the left most child in cases of non binary b-trees) until we are on a leaf, then we return the first key.

Algorithm 1: Tree Maximum

```
1 B_TREE_FIND_MIN(x):
2       while (x.first_child != NULL):
3           x == x.first_child
4       end
5       return x.key[0]
```

To find the predecessor of a given key, first we need to find the given key. If the given key is on a leaf then we return the preceding key. If the key is not in a leaf node, then we return the largest element of the child that immediately precedes the key.

Algorithm 2: Tree Maximum

```
1 FIND_PRECEEDING_KEY(x, k):
2       x, i = SEARCH_KEY(x, k)
3       if k == B_TREE_FIND_MIN(x):
4           return NULL
5       end
6       else if x.child == NULL:  # check if x is a leaf node
7           return x[i - 1]
8       end
9       else:
10          MAX(x.left_child[i])
```

Problem 2

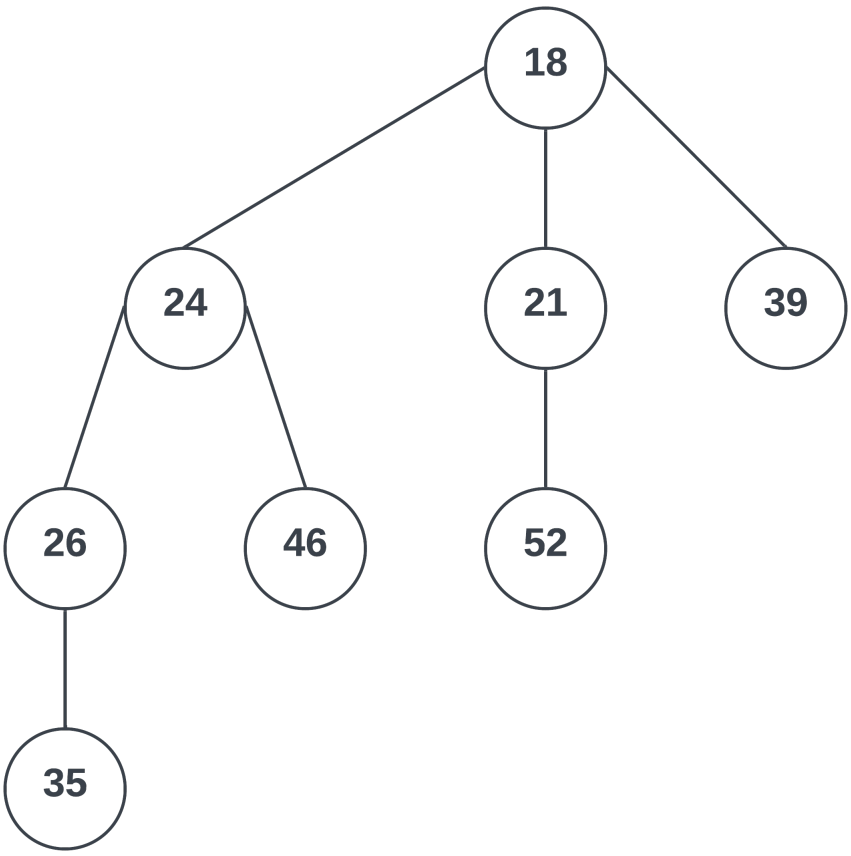

Figure 1: Fibonacci heap result

Problem 3

- The path required can be obtained from the order in which DFS explores the edges.
- When traversing an edge that goes to an unvisited node the edge $(u- > v)$ is added in the path.
- When we backtrack to u again after v is made $BLACK$, the edge (v -¿ u) is added to the path.
- In this way each edge is added to the traversal path exactly once in each direction.

Problem 4

- Assume we have a minimal spanning tree T that contains a light edge (u, v) that doesn't cross any graph cuts.
- Then there must exist another edge (x, y) with smaller weight than (u, v) that separates u and v.
- Now lets consider a cut $(S, V - S)$ that separates u and v.
- There is a path from u to v in T that crosses the cut at some point.
- Let (x, y) be the first edge on this path that crosses the cut
- Since (x, y) crosses the cut $(S, V - S)$, we have $w(x, y) < w(u, v)$
- We replace (u, v) with (x, y). This creates a new tree T_2
- Since (x, y) is in T, T_2 is also a spanning tree of the graph.
- But $w(x, y)$ is less than $w(u, v)$ i.e. T_2 is lighter than T.
- This contradicts are assumption that T is a minimal spanning tree
- Therefore, if an edge (u, v) is in a minimal spanning tree, it must cross a graph cut as a light edge.