# Homework Assignment 8

**Due Date:** April 18, 2021, 23:59

**Note.** Please note that this semester all assignments are group assignments. Further note that for the grading we will apply a "10%" rule, i.e. the maximum number of points for this assignments is 55, but 50 will be counted as 100%. Points that exceed 50 will be stored in a separate counter and used later for compensation of lost points in other assignments or (if not used up this way) the final exam.

Exercise 1.

- **(i)** Explain how to find the minimum and the maximum key stored in a B-tree.
- **(ii)** Implement operations on B-trees to return the record associated with the minimum and maximum keys, respectively.
- **(iii)** Explain how to find the predecessor and the successor keys of a given key stored in a B-tree.
- **(iv)** Implement operations on B-trees to return the record associated with the predecessor and the successor of a given key.

**total points: 15**

Exercise 2.

- **(i)** Explain how to find the minimum and the maximum key stored in a $B^+$-tree.
- **(ii)** Implement operations on $B^+$-trees to return the record associated with the minimum and maximum keys, respectively.
- **(iii)** Explain how to find the predecessor and the successor keys of a given key stored in a $B^+$-tree.
- **(iv)** Implement operations on $B^+$-trees to return the record associated with the predecessor and the successor of a given key.

**total points: 15**

Exercise 3. For the bipartite matching problem we are given a finite bipartite graph $(V, E)$, where the set $V$ of vertices is partitioned into two sets *Boys* and *Girls* of equal size. Thus, the set $E$ of edges contains sets $\{x, y\}$ with $x \in \mathit{Boys}$ and $y \in \mathit{Girls}$. A *perfect matching* is a subset $F \subseteq E$ such that every vertex is incident to exactly one edge in $F$. A *partial matching* is a subset $F \subseteq E$ such that every vertex is incident to at most one edge in $F$. So the algorithm will create larger and larger partial matchings until no more unmatched boys and girls are left, otherwise no perfect matching exists.

We use functions girls_to_boys and boys_to_girls turning sets of unordered edges into sets of ordered pairs:

$$\text{girls\_to\_boys}(X) = \{(g,b) \mid b \in \mathit{Boys} \wedge g \in \mathit{Girls} : \{b,g\} \in X\}$$
$$\text{boys\_to\_girls}(X) = \{(b,g) \mid b \in \mathit{Boys} \wedge g \in \mathit{Girls} : \{b,g\} \in X\}$$

Conversely, the function unordered turns a set of ordered pairs $(b,g)$ or $(g,b)$ into a set of two-element sets:

$$\text{unordered}(X) = \{\{x,y\} \mid (x,y) \in X\}$$

We further use a predicate reachable and a function path. For the former one we have reachable$(b,X,g)$ iff there is a path from $b$ to $g$ using the directed edges in $X$. For the latter one path$(b,X,g)$ is a set of ordered pairs representing a path from $b$ to $g$ using the directed edges in $X$.

Then an algorithm for bipartite matching can be realised by iterating the following rule:

**par** **if** $\mathit{mode} = \text{init}$
    **then** **par** $\mathit{mode} := \text{examine}$
            $\mathit{partial\_match} := \emptyset$
        **endpar**
    **endif**
    **if** $\mathit{mode} = \text{examine}$
    **then** **if** $\exists b \in \mathit{Boys}.\forall g \in \mathit{Girls}.\{b,g\} \notin \mathit{partial\_match}$
        **then** $\mathit{mode} := \text{build-digraph}$
        **else** **par** $\mathit{Output} := \mathbf{true}$
                $\mathit{Halt} := \mathbf{true}$
                $\mathit{mode} := \text{final}$
            **endpar**
        **endif**
    **endif**
    **if** $\mathit{mode} = \text{build-digraph}$
    **then** **par** $\mathit{di\_graph} := \text{girls\_to\_boys}(\mathit{partial\_match})$
                $\cup\ \text{boys\_to\_girls}(E - \mathit{partial\_match})$
            $\mathit{mode} := \text{build-path}$
        **endpar**
    **endif**
    **if** $\mathit{mode} = \text{build-path}$
    **then choose** $b \in \{x \mid x \in \mathit{Boys} : \forall g \in \mathit{Girls}.\{b,g\} \notin \mathit{partial\_match}\}$
        **do** **if** $\exists g' \in \mathit{Girls}.\forall b' \in \mathit{Boys}.\{b',g'\} \notin \mathit{partial\_match}$
                $\wedge\ \text{reachable}(b, \mathit{di\_graph}, g')$
            **then choose** $g \in \{y \mid y \in \mathit{Girls}.\forall x \in \mathit{Boys}.\{x,y\}$
                $\notin \mathit{partial\_match} \wedge\ \text{reachable}(b, \mathit{di\_graph}, y)\}$
                **do** **par** $\mathit{path} := \text{path}(b, \mathit{di\_graph}, g)$
                        $\mathit{mode} := \text{modify}$
                    **endpar**
                **enddo**
            **else** **par** $\mathit{Output} := \mathbf{false}$

```
                        Halt := true
                        mode := final
                    endpar
                endif
            enddo
        endif
        if      mode = modify
        then    par     partial_match = (partial_match − unordered(path))
                                        ∪(unordered(path) − partial_match)
                        mode := examine
                endpar
        endif
endpar
```

**(i)** Implement the above algorithm for the determination of a perfect matching on a bipartite graph (provided such a matching exists).

**total points: 12**

Exercise 4.

**(i)** Implement a class BiPartiteGraph of bipartite graphs covering basic operations for insertion and deletion of vertices and edges and for the determination of edges incident to a given vertex.

**(ii)** Implement a program that determines, whether a perfect matching exists for a given bipartite graph (not the algorithm from the previous exercise).

**total points: 13**