Technische Universität München
Department of Computer Science
Chair for Efficient Algorithms
Prof. Dr. Sami Khuri

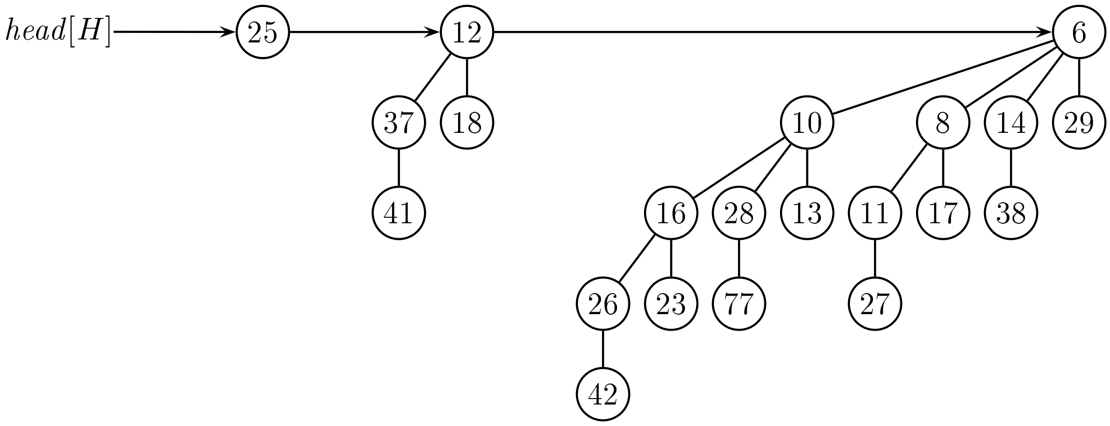
WS 1999/2000
Problem Set 8
December 21, 1999

Efficient Algorithms and Data Structures

Note: Solutions for this problem set are due on January 11, 2000.

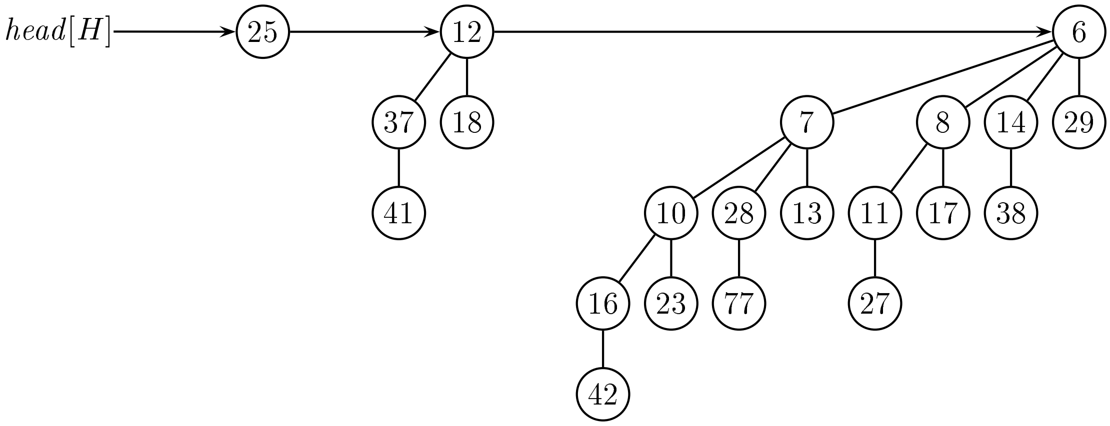
Problem 1 (6 credits)

(Exercise 20.2-3 in [CLR90, p. 417]) Show the binomial heap that results when a node with key 24 is inserted into this binomial heap:



Problem 2 (11 credits)

(Exercise 20.2-4 in [CLR90, p. 417]) Show the binomial heap that results when the node with key 28 is deleted from this binomial heap:



Problem 3 (12 credits)

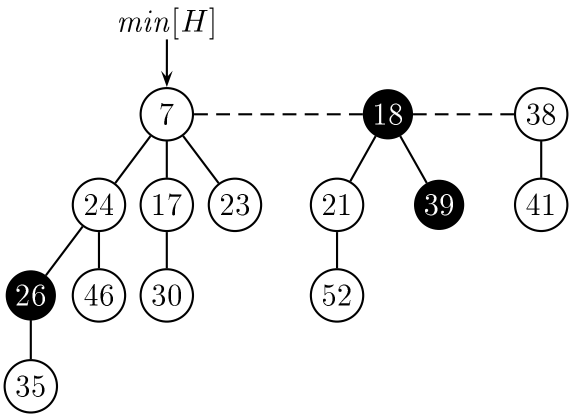
(Problem 20-2 in [CLR90, pp. 418–419]) We are given a connected, undirected graph $G = (V, E)$ with a weight function $w : E \to \mathbb{R}$. $w(u, v)$ is called the weight of edge (u, v). We wish to find a minimum spanning tree for G, i.e., a spanning tree $T \subseteq E$ such that $w(T) = \sum_{(u,v)\in T} w(u, v)$ is minimized. We use mergeable heaps to construct a minimum-spanning-tree algorithm different from Prim's algorithm and Kruskal's algorithm. Consider the following pseudocode, which maintains a partition $\{V_i\}$ of the vertices of V and, with each set V_i, a set $E_i \subseteq \{(u, v) : u \in V_i \text{ or } v \in V_i\}$ of edges incident on vertices in V_i.

```
MST-MERGEABLE-HEAP(G)
T ← ∅
for each vertex v_i ∈ V[G]
     do V_i ← {v_i}
          E_i ← {(v_i, v) ∈ E[G] }
while there is more than one set V_i
     do choose any set V_i
          extract the minimum-weight edge (u, v) from E_i
          assume without loss of generality that u ∈ V_i and v ∈ V_j
          if i ≠ j
             then T ← T ∪ {(u, v)}
                    V_i ← V_i ∪ V_j, destroying V_j
                    E_i ← E_i ∪ E_j
```

Describe how to implement this algorithm using the mergeable-heap operations, i.e., the operations MAKE-HEAP, INSERT, MINIMUM, EXTRACT-MIN, and UNION. Give the running-time of your implementation, assuming that the mergeable heaps are implemented by binomial heaps.

Problem 4 (11 credits)

(Exercise 21.2-1 in [CLR90, p. 431]) Show the Fibonacci heap that results from calling FIB-HEAP-EXTRACT-MIN on this Fibonacci heap (marked nodes are shown black):



References

[CLR90] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. MIT Press, Cambridge, 1990.