Algorithms (COT 6405): Solutions 6

Problem 1

A *d-ary heap* is like a binary heap, but instead of 2 children, nodes have d children.

(a) How would you represent a d-ary heap with n elements in an array? What are the expressions for determining the parent of a given element, $\text{PARENT}(i)$, and a j-th child of a given element, $\text{CHILD}(i, j)$, where $1 \le j \le d$?

The following expressions determine the parent and j-th child of element i (where $1 \le j \le d$):

$$\begin{aligned} \text{PARENT}(i) &= \left\lfloor \frac{i + d - 2}{d} \right\rfloor, \\ \text{CHILD}(i, j) &= (i - 1) \cdot d + j + 1. \end{aligned}$$

(b) Write an efficient implementation of HEAPIFY and HEAP-INSERT for a d-ary heap.

The HEAPIFY algorithm is somewhat different from the binary-heap version, whereas HEAP-INSERT is identical to the corresponding algorithm for binary heaps. The running time of HEAPIFY is $O(d \cdot \log_d n)$, and the running time of HEAP-INSERT is $O(\log_d n)$.

```
HEAPIFY(A, i, n, d)
largest ← i
for l ← CHILD(i, 1) to CHILD(i, d)        ▷ loop through all children of i
    do  if l ≤ n and A[l] > A[largest]
             then largest ← l
if largest ≠ i
    then exchange A[i] ↔ A[largest]
         HEAPIFY(A, largest)
```

```
HEAP-INSERT(A, key)
heap-size[A] ← heap-size[A] + 1
i ← heap-size[A]
while i > 1 and A[PARENT(i)] < key
     do A[i] ← A[PARENT(i)]
        i ← PARENT(i)
A[i] ← key
```

Problem 2

What is the height of a d-ary heap of n elements in terms of n and d?

The height h of a heap is *approximately* equal to $\log_d n$. The exact height is

$$h = \lceil \log_d(n \cdot d - n + 1) - 1 \rceil.$$