Datastructuren

October 26, 2009

1 Heaps

1. Consider the following lists of integers: $L_1 = (1, 3, 1, 6, 9, 7, 12, 8, 7, 9, 10, 10)$ and $L_2 = (2, 6, 12, 4, 9, 8, 10, 7, 15, 3)$. Are these lists in a min-heap configuration? If not, what changes are needed to turn it into a min-heap configurations? (Convention: the node which is represented in position i has its children (if any) in positions $2i$ and $2i + 1$.)

2. Carry out the min-heap insertion of the element 2 on the list L_1 of Exercise 1.

3. Does a list which is sorted in ascending order represent a min-heap?

4. Is a min-heap always represented by a list which is sorted in ascending order?

5. Where is the minimum element of a min-heap located?

6. Where is the maximum element of a min-heap located?

7. What is the height h of a min-heap consisting of n elements?

8. Write a function *isALeaf*(i, H) where H is a min-heap and i is an integer which returns a boolean equal to "true", if the index i corresponds to a leaf of H. Write a function *existsRightChild*(i, H) which returns a boolean "true" if the node associated to the index i of the heap H has a right child.

9. Write the functions *leftChild*(i, H), *rightChild*(i, H) and *parent*(i, H) which return the index of the left child, the right child and of the parent respectively of the node at index i of the heap H.

10. Write a procedure *insert*(e, H) which allows for inserting an element e into a min-heap H; here we assume that the elements are integers with the usual ordering. (Or more generally: allow for a priority function p to be defined on the set of elements E and thus the input parameters are e, p, H, where $p : E \to \mathbb{Z}$; now the heap condition becomes: $\forall$ nodes x, in case the right child, denoted by x_R exists we have $p(\sigma(x)) < p(\sigma(x_R))$ and in case the left child exists, denoted by x_L, we have $p(\sigma(x)) < p(\sigma(x_L))$. The function σ maps the nodes to elements in E.) What is the complexity of this operation?.

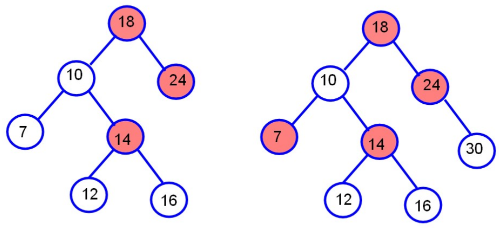

Figure 1: red black trees

11. Write a procedure *deleteMin*(H) which deletes the smallest element of the min-heap H. What is its complexity?

12. a Neem aan dat het array $A[1..N]$ gevuld is met gehele getallen. Er zijn twee voor de hand liggende manieren om dit array tot een heap om te zetten.

    Bij de eerste manier werken we top-down: Als $A[1..i-1]$ al een heapstructuur heeft, voegen we $A[i]$ toe en borrelen we dit element naar de juiste plaats. De tweede aanpak is bottom-up. We voegen telkens $A[i]$ samen met de twee heaps gevormd door de kinderen $A[2i]$ en $A[2i+1]$ met al hun nakomelingen. Nu zakt $A[i]$ juist naar beneden.

    a Werk deze twee algoritmen verder uit tot pseudo-code of C++ functies.

    b Bespreek de efficientie van de methoden. Welke manier heeft de voorkeur?

13. (‡) Onderzoek hoe twee *heaps* gebruikt kunnen worden om van een verzameling snel zowel het grootste als het kleinste element te kunnen vinden (en verwijderen), terwijl nieuwe elementen in logaritmische tijd toegevoegd kunnen worden. Gebruik één van de heaps als *maxheap* (elke knoop heeft een sleutel die niet groter is dan die van zijn ouder) de ander als *minheap* (idem, maar sleutels niet kleiner dan die van ouder). Zorg ervoor dat tijdens het gebruik beide heaps ongeveer evenveel elementen blijven bevatten.

2 Red-Black Trees

1. Among the trees in Figure 1 which of them is a balanced red-black tree?

2. Construct the Red-Black tree starting from an empty tree and inserting the following keys in the order given: 4, 7, 3, 2, 5, 6.

3. a Wat zijn de kenmerkende eigenschappen van de rood-zwart boom (red-black tree)?

   b Gegeven is de rood-zwart boom in Figure 2, waarbij 'rode' knopen een extra cirkel hebben gekregen. Welke boom onstaat als we hieraan toevoegen

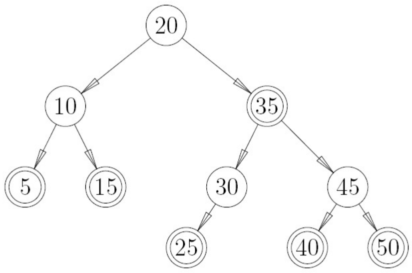

Figure 2: red black trees

i. eerst 55 en vervolgens 7.
ii. eerst 7 en vervolgens 55.

Benoem de achtervolgende operaties en geef relevante tussenresultaten.

c Hoeveel knopen (rode èn zwarte samen) heeft een rood-zwarte boom van hoogte h maximaal? Minimaal?

De hoogte van de boom wordt hier gemeten in het aantal knopen op het langste pad van wortel naar blad.

3 B-trees

Determine an upper bound for the height of a B-tree or order m.