מטלה 13 – מבני נתונים ומבוא לאלגוריתמים

מגיש : אלון אלמוג

(הערה : פתרתי את סעיף ב' קודם מטעם נוחיות, ובנוסף נעזרתי בו כדי לפתור את סעיף א'.)

בעיה 6-2 סעיף ב' :

עבור עץ המייצג את הערימה הd-ית, נסמן את מספר האיברים בעץ ב – n (שזהו מספר האיברים בערימה) את גובה העץ ב-h, ואת הרמה המלאה האחרונה ב-k.

במקרה שהעץ מלא, הרמה ה-k ית היא הרמה האחרונה בעץ, לכן מספר האיברים בעץ שווה לסכום איברי כל הרמות בעץ. אחרת, אם ישנה רמה אחרי k (ברור שאינה מלאה מהגדרת k), אם רמה זו הייתה מלאה, בהכרח היו בה יותר מ-n איברים (מאחר ואלו כל איברי העץ הלא מלא, שסכומם בוודאי קטן מסכום איברי העץ אילו הרמה ה-k+1 הייתה מלאה).

לכן נקבל :

לפי הנוסחה לסכום של סדרה הנדסית1 –

כעת נפעיל logd ונקבל –

לכן לפי תכונות ערך תקרה2 נקבל , וזהו גובה העץ.

בעיה 6-2 סעיף א' :

יהי האינדקס i של איבר בעץ (המייצג כמובן ערימה). בכדי למצוא את האינדקסים של בניו, ראשית אסכום כמה איברים יש בעץ עד הרמה בה נמצא האינדקס. לאחר מכן אספור כמה איברים נמצאים ברמה של האינדקס, כך שהאינדקס שלהם קטן ממנו (כלומר כמה איברים באותה הרמה נמצאים לפניו). לכל אחד מהאינדקסים האלה, ישנם d בנים, לכן אם אספור כמה d-ים יש סה"כ לכל האינדקסים ברמה בה i נמצא, ולכל אחד מהם אספור d בנים, אגיע לכך ש-d האינדקסים הבאים הם הבנים של האינדקס i המבוקש.

כמות האיברים עד הרמה שבה i נמצא :

וכמות האיברים הנמצאים ברמה של i עד i עצמו (לא כולל) :

לכן לפי ההסבר שכתבתי קודם, המשוואה הסופית למציאת הבן ה-k-י של i :

כעת, בכדי למצוא אב של בן מסויים, הנוסחה שהגענו אליה תקפה גם למציאת האב. נסמן i = p (מלשון parent), ונסמן את s (מלשון son) בתור האינדקס המייצג את הבן שאת אביו אנחנו מחפשים.

נקבל לפי הנוסחה :

ולפי תכונות ערך תקרה2 נקבל כי

וסיימנו.

1 ויקיפדיה – סדרה הנדסית, חלק 4 'סימון חלופי'.

2 wikipedia - floor and ceiling functions, part 2 ‘equivalences’.

In this program i’ve implemented a maximum heap, in which each parent is equal or bigger than any of his sons, using an array.

How to use the program :

* enter manually (via keyboard) the value of d.
* enter one of the following commands **without whitespace at the end of each command**.

Documentation

Variables :

* int “D” represents the value of the chosen d order for the d-ary heap.
* int “heapSize” represents the size of the heap at every given moment, and updates each time the heap is being expanded and contracted.
* int[] “Heap” is the array with which the heap is being implemented.

Methods :

* “checkValid()” checks in the chosen d is a natural number that’s bigger than 1. If that is not the case, the method asks the user to try again.
* “parent(int i)” returns the index of the parent of a given index (given index is ‘i’, and while ‘i’ isn’t the first element in the heap).
* “maxHeapify(int[] arr, int i)” in a heap where only ‘i’ is an index that contains an element that disturbs the heap (max) property, the method modifies the heap so it obtains the heap (max) property.
* “extractMax(int[] arr)” extract the biggest element in the heap (which is the first element) and modifies the rest of the heap so it preserves the heap property.
* “increaseKey(int[] arr, int i, int key)” increases a key’s value to that of key (in case that original key’s value is bigger than the key parameter, returns “new key is smalled than current key”.
* “insert(int[] arr, int key)” inserts a key in the next open space of the array that represents the heap. if the array is already full, returns “heap is already full!”.
* “delete(int[] arr, int i)” deletes the element who’s index it ‘i’. if the heap is empty, returns “heap is already empty”.
* “print(int[] arr, level)” prints the heap. How to read : For each row, the sons of the first element are the first d elements in the lower row. the sons of the second element are the next d elements after that, and so on.

Analyzing Time Complexity :

max-heapify() – as proves in class (exactly like is written in the presentation), the time complexity of this method is Θ(height of tree rooted in index ‘i’), which in turn is (as proven in 6-2 b) Θ(.

extract-max() – the time complexity of this method is the same as max-heapify(), which is Θ(.

increase-key(i, k) – In the worst case, the key will be promoted until it reaches the first element and be placed in it’s place. there for it is the height of the tree that’s representing the heap, which is Θ(

insert(k) – the time complexity of this method is the same as increase-key(i, k), which means it’s Θ(

delete(i) – The method calls the methods increase-key(i, k) and extract-max(), which means it’s time complexity is Θ(Θ( = Θ(.

print() – The method calls to itself times, and each call’s time complexity is equal to the number of elements in the current level, which means :

which means the time complexity of the method is (nd-n+1)\*Θ(, which is Θ(

tester outputs :

output for tester1d2 –

Welcome to 'Build a D-th Heap' Program!

Please enter the value of D

2

The value of d is : 2

The function increase-key(0, 23) has been executed successfuly.

The function insert-key(23) has been executed successfuly.

The function increase-key(1, 45) has been executed successfuly.

The function insert-key(45) has been executed successfuly.

The function increase-key(2, 34) has been executed successfuly.

The function insert-key(34) has been executed successfuly.

45

23|34

The function increase-key(0, 50) has been executed successfuly.

The function increase-key(3, 30) has been executed successfuly.

The function insert-key(30) has been executed successfuly.

The function increase-key(4, 10) has been executed successfuly.

The function insert-key(10) has been executed successfuly.

The function increase-key(5, 4) has been executed successfuly.

The function insert-key(4) has been executed successfuly.

50

30|34

23|10|4

The function extract-max() has been executed successfuly.

34

30|4

23|10

The function increase-key(0, 35) has been executed successfuly.

The function extract-max() has been executed successfuly.

Key has been deleted successfully.

30

23|4

10

The function increase-key(4, 32) has been executed successfuly.

The function insert-key(32) has been executed successfuly.

The function increase-key(5, 12) has been executed successfuly.

The function insert-key(12) has been executed successfuly.

The function increase-key(6, 22) has been executed successfuly.

The function insert-key(22) has been executed successfuly.

The function increase-key(7, 31) has been executed successfuly.

The function insert-key(31) has been executed successfuly.

The function increase-key(8, 7) has been executed successfuly.

The function insert-key(7) has been executed successfuly.

The function increase-key(9, 3) has been executed successfuly.

The function insert-key(3) has been executed successfuly.

The function increase-key(10, 15) has been executed successfuly.

The function insert-key(15) has been executed successfuly.

The function increase-key(11, 19) has been executed successfuly.

The function insert-key(19) has been executed successfuly.

The function increase-key(12, 15) has been executed successfuly.

The function insert-key(15) has been executed successfuly.

32

31|22

30|23|19|12

10|7|3|15|4|15

The function increase-key(12, 33) has been executed successfuly.

33

31|32

30|23|22|12

10|7|3|15|4|19

The function increase-key(0, 34) has been executed successfuly.

The function extract-max() has been executed successfuly.

Key has been deleted successfully.

32

31|22

30|23|19|12

10|7|3|15|4

The function extract-max() has been executed successfuly.

31

30|22

10|23|19|12

4|7|3|15

The function increase-key(11, 2) has been executed successfuly.

The function insert-key(2) has been executed successfuly.

The function increase-key(12, 1) has been executed successfuly.

The function insert-key(1) has been executed successfuly.

The function increase-key(13, 6) has been executed successfuly.

The function insert-key(6) has been executed successfuly.

The function increase-key(14, 5) has been executed successfuly.

The function insert-key(5) has been executed successfuly.

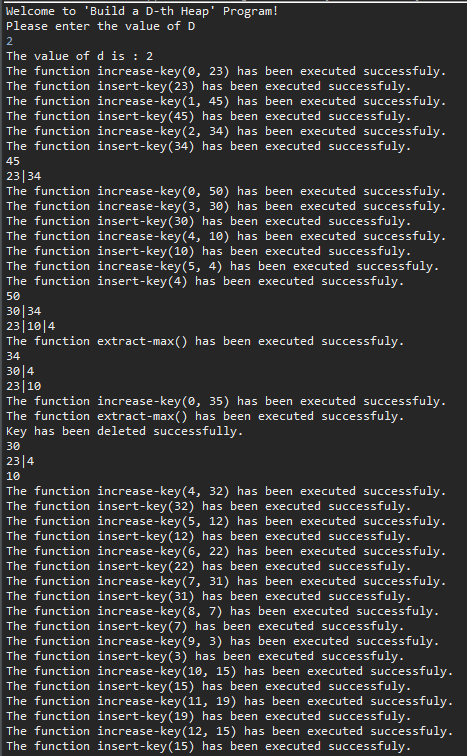
31

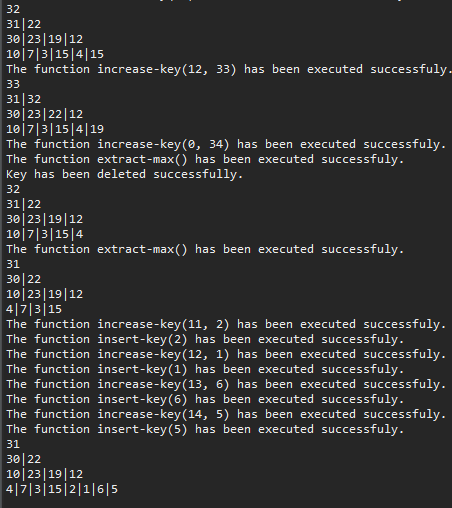
30|22

10|23|19|12

4|7|3|15|2|1|6|5

and as a picture :





output for tester2d3 –

Welcome to 'Build a D-th Heap' Program!

Please enter the value of D

3

The value of d is : 3

The function increase-key(0, 1) has been executed successfuly.

The function insert-key(1) has been executed successfuly.

The function increase-key(1, 2) has been executed successfuly.

The function insert-key(2) has been executed successfuly.

The function increase-key(2, 3) has been executed successfuly.

The function insert-key(3) has been executed successfuly.

The function increase-key(3, 4) has been executed successfuly.

The function insert-key(4) has been executed successfuly.

The function increase-key(4, 5) has been executed successfuly.

The function insert-key(5) has been executed successfuly.

The function increase-key(5, 6) has been executed successfuly.

The function insert-key(6) has been executed successfuly.

The function increase-key(6, 7) has been executed successfuly.

The function insert-key(7) has been executed successfuly.

The function increase-key(7, 8) has been executed successfuly.

The function insert-key(8) has been executed successfuly.

The function increase-key(8, 9) has been executed successfuly.

The function insert-key(9) has been executed successfuly.

The function increase-key(9, 10) has been executed successfuly.

The function insert-key(10) has been executed successfuly.

The function increase-key(10, 1) has been executed successfuly.

The function insert-key(1) has been executed successfuly.

The function increase-key(11, 2) has been executed successfuly.

The function insert-key(2) has been executed successfuly.

The function increase-key(12, 3) has been executed successfuly.

The function insert-key(3) has been executed successfuly.

The function increase-key(13, 4) has been executed successfuly.

The function insert-key(4) has been executed successfuly.

The function increase-key(14, 5) has been executed successfuly.

The function insert-key(5) has been executed successfuly.

The function increase-key(15, 6) has been executed successfuly.

The function insert-key(6) has been executed successfuly.

The function increase-key(16, 7) has been executed successfuly.

The function insert-key(7) has been executed successfuly.

The function increase-key(17, 8) has been executed successfuly.

The function insert-key(8) has been executed successfuly.

The function increase-key(18, 9) has been executed successfuly.

The function insert-key(9) has been executed successfuly.

The function increase-key(19, 10) has been executed successfuly.

The function insert-key(10) has been executed successfuly.

The function increase-key(20, 1) has been executed successfuly.

The function insert-key(1) has been executed successfuly.

The function increase-key(21, 2) has been executed successfuly.

The function insert-key(2) has been executed successfuly.

The function increase-key(22, 3) has been executed successfuly.

The function insert-key(3) has been executed successfuly.

The function increase-key(23, 4) has been executed successfuly.

The function insert-key(4) has been executed successfuly.

The function increase-key(24, 5) has been executed successfuly.

The function insert-key(5) has been executed successfuly.

The function increase-key(25, 6) has been executed successfuly.

The function insert-key(6) has been executed successfuly.

The function increase-key(26, 7) has been executed successfuly.

The function insert-key(7) has been executed successfuly.

The function increase-key(27, 8) has been executed successfuly.

The function insert-key(8) has been executed successfuly.

The function increase-key(28, 9) has been executed successfuly.

The function insert-key(9) has been executed successfuly.

The function increase-key(29, 10) has been executed successfuly.

The function insert-key(10) has been executed successfuly.

The function increase-key(30, 1) has been executed successfuly.

The function insert-key(1) has been executed successfuly.

The function increase-key(31, 2) has been executed successfuly.

The function insert-key(2) has been executed successfuly.

The function increase-key(32, 3) has been executed successfuly.

The function insert-key(3) has been executed successfuly.

The function increase-key(33, 4) has been executed successfuly.

The function insert-key(4) has been executed successfuly.

The function increase-key(34, 5) has been executed successfuly.

The function insert-key(5) has been executed successfuly.

The function increase-key(35, 6) has been executed successfuly.

The function insert-key(6) has been executed successfuly.

The function increase-key(36, 7) has been executed successfuly.

The function insert-key(7) has been executed successfuly.

The function increase-key(37, 8) has been executed successfuly.

The function insert-key(8) has been executed successfuly.

The function increase-key(38, 9) has been executed successfuly.

The function insert-key(9) has been executed successfuly.

The function increase-key(39, 10) has been executed successfuly.

The function insert-key(10) has been executed successfuly.

10

10|10|10

6|8|9|5|8|9|3|6|9

1|4|5|4|6|7|5|1|2|2|3|4|6|7|7|8|9|1|1|2|3|2|4|5|3|7|8

The function increase-key(40, 2) has been executed successfuly.

The function insert-key(2) has been executed successfuly.

The function increase-key(0, 100) has been executed successfuly.

100

10|10|10

6|8|9|5|8|9|3|6|9

2|4|5|4|6|7|5|1|2|2|3|4|6|7|7|8|9|1|1|2|3|2|4|5|3|7|8

1

The function extract-max() has been executed successfuly.

10

9|10|10

6|8|5|5|8|9|3|6|9

2|4|5|4|6|7|1|1|2|2|3|4|6|7|7|8|9|1|1|2|3|2|4|5|3|7|8

and as a picture :

