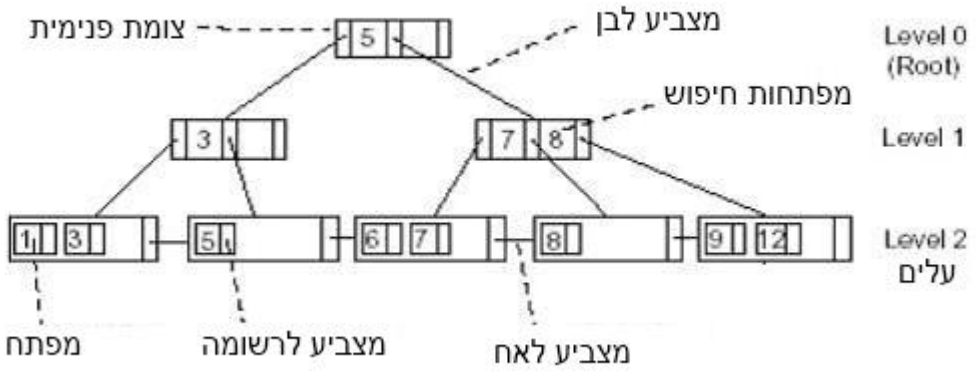
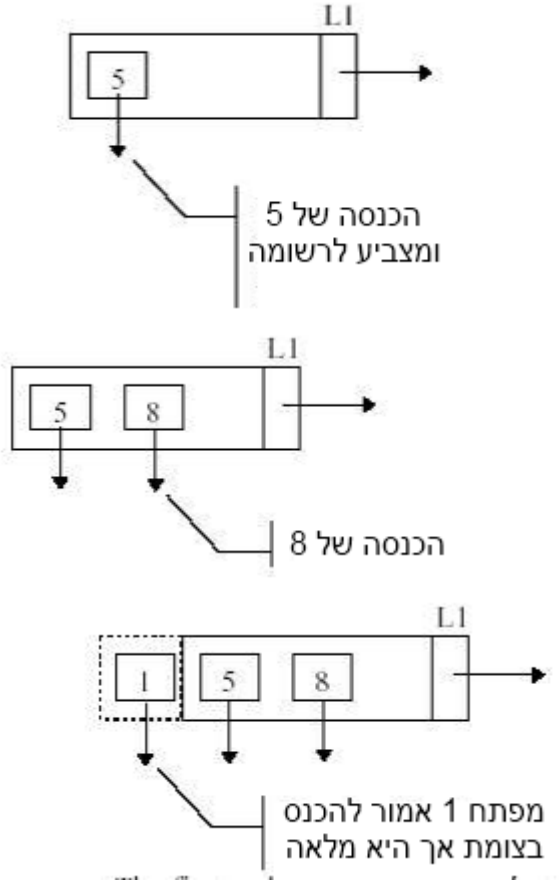
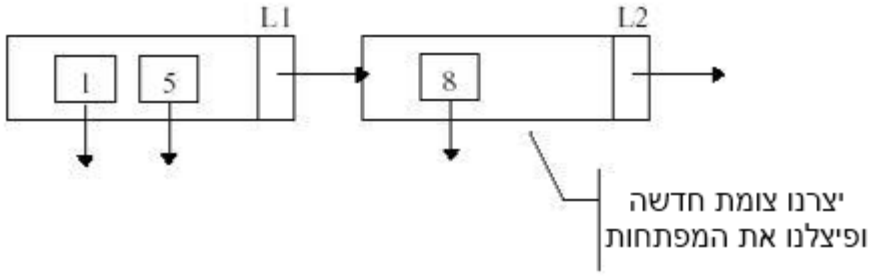
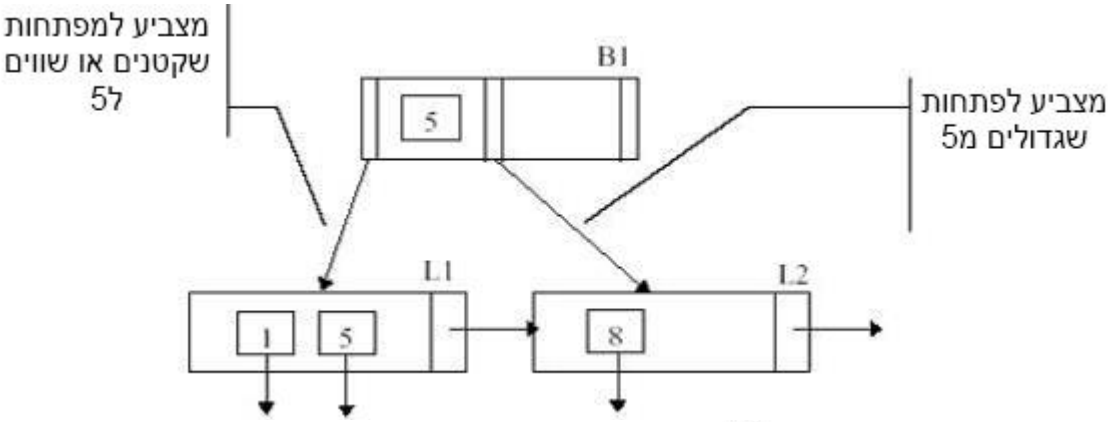
Implementation of a B+ tree.

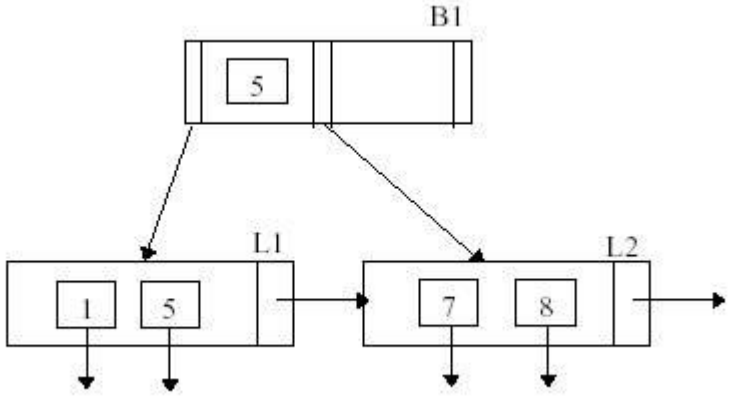


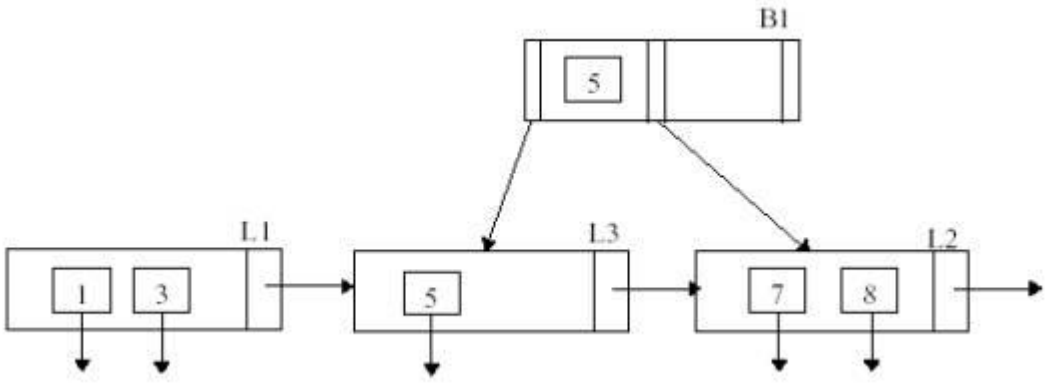
Example for t=3: 5, 8, 1, 7, 3, 12, 9, 6

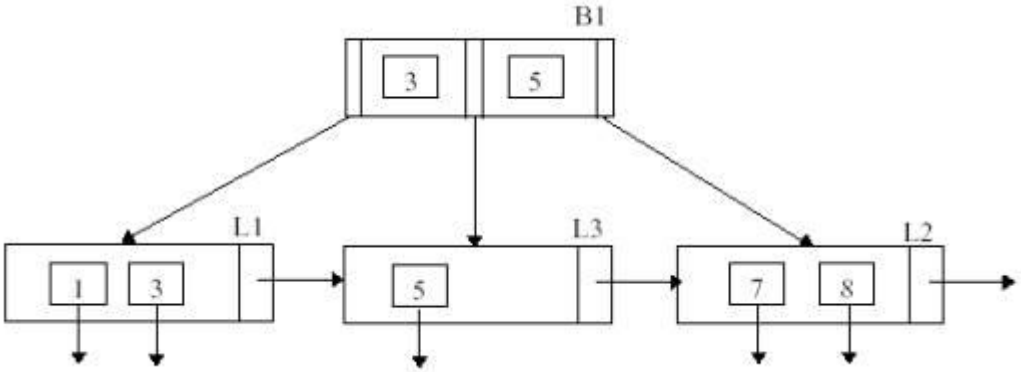


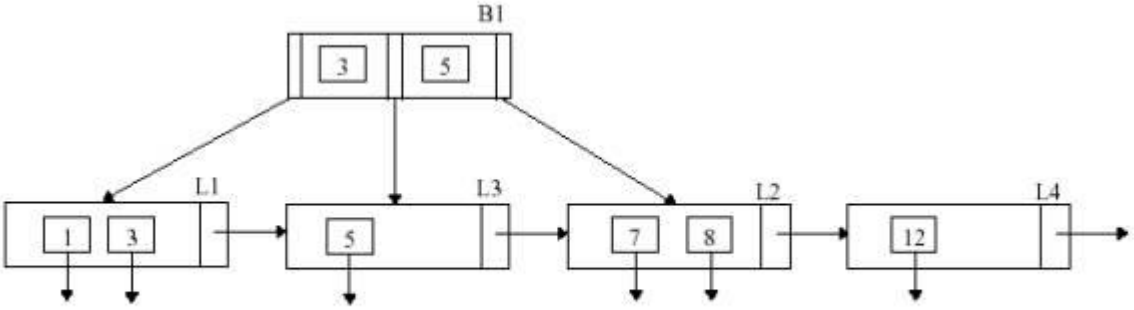

L1
L2
1
5
8
יצרנו צומת חדשה
ופיצלנו את המפתחות

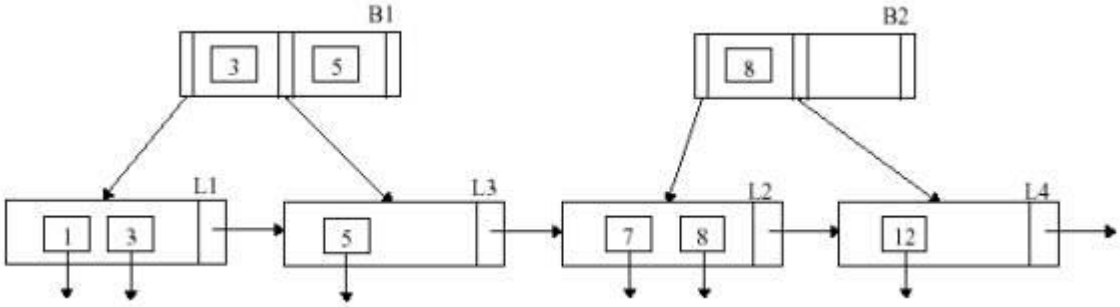

מצביע למפתחות
שקטנים או שווים
ל5
B1
5
מצביע לפתחות
שגדולים מ5
L1
L2
1
5
8


B1
5
L1
L2
1
5
7
8

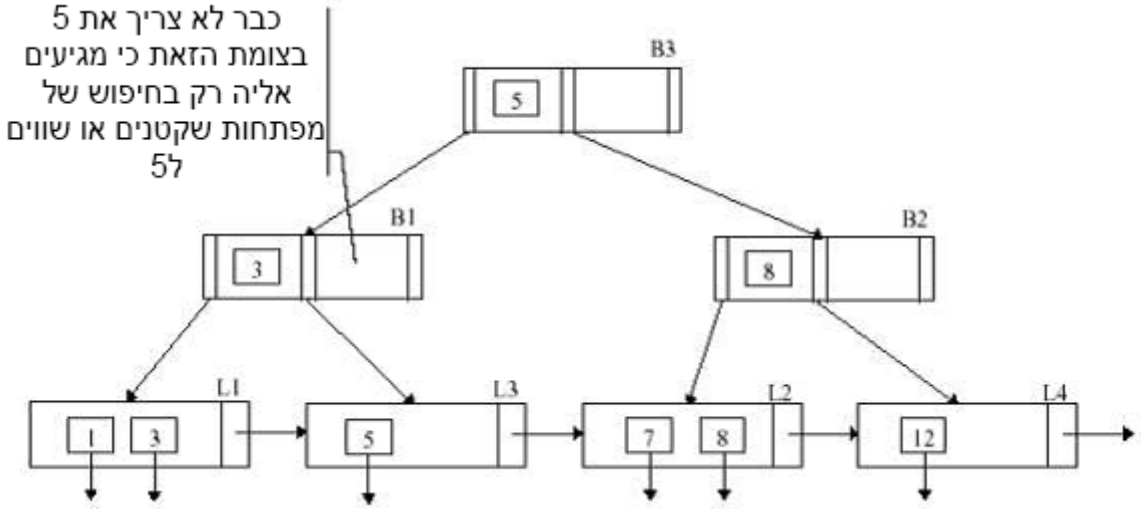

B1
5
L1
L3
L2
1
3
5
7
8

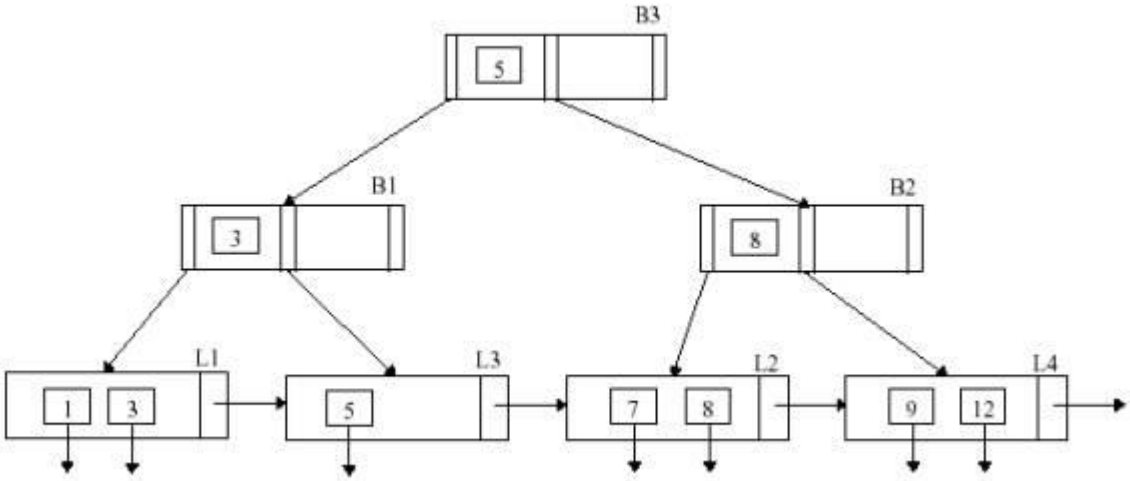

B1
3
5
L1
L3
L2
1
3
5
7
8

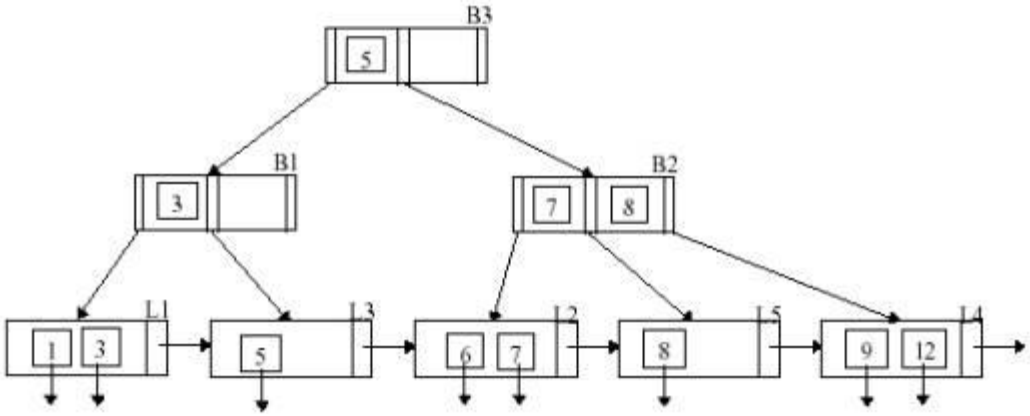

B1
3
5
L1
L3
L2
L4
1
3
5
7
8
12


B1
B2
3
5
8
L1
L3
L2
L4
1
3
5
7
8
12

כבר לא צריך את 5
בצומת הזאת כי מגיעים
אליה רק בחיפוש של
מפתחות שקטנים או שווים
ל5







Requirements:

Design a data structure that supports the following procedures in the given run time:

Procedure	Description	Run time
Init(t)	Initialize with parameter t	$O(1)$
Insert(x)	Insert x	$O(\log_t n)$
Search(x)	Find x	If x exists: $O(\log_t n)$ If x doesn't exist: $O(1)$ (high probability)
Min_gap()	Find the smallest difference between 2 numbers in the structure	$O(1)$
Order(x)	Finds the location of x as if the structure is sorted from the smallest to largest number	$O(\log_t n)$

All tree members are natural numbers.

Input/Output:

Input:

The program will run from the command line, with 4 arguments:
1. Input file name
2. Parameter t
3. A number from the file (is used for the 'order' procedure)
4. Output file name

Example:
java **BPlus** input.dat 3 7 output1.dat

The input file will contain natural numbers divided by spaces, which the program will enter in that order. The second parameter is the tree's constant. The third is one of the numbers from the input file, and the forth is the name of the output file.
*It is safe to assume all given numbers are different

Output:

The output file will show all the tree's leaves in order, leftmost to rightmost. The members of each leaf will be divided by commas. Leaves will be notated by a hash tag. After the tree is printed, the minimal tree gap will be printed, and the order(x) of the third argument given to the program.

Example:
java BPlus input.dat 3 7 output1.dat
With input file containing the example given at the beginning of this document,
will return an output1.dat file, containing:

1,3#5#6,7#8#9,12– leaves divided by hash tags
1- the minimal gap in the tree (between 5 and 6)

5- the order of the number 7