Last name	
First name	
Group	

Grade	

Algorithmics
S3
Mid-term Exam #3
26 oct. 2015 - 14:00
(D.S. 308818.03 BW)
Answer Sheets

I	
II	
III	

Answers 1 (**Some questions** − *5 points*)

1. Give 2 required properties of a hash function.

    (a) Property 1: _____

    (b) Property 2: _____

2. Give a direct method of hashing:

    _____

3. Give a indirect method of hashing:

    _____

4. Which collision resolution method does not need a hash table whose size is greater than the number of keys to be hashed ?

    _____

5. Which kind of search is incompatible with the hashing?

    _____

6. With which collision resolution method do secondary collisions appear?

    _____

Answers 2 **(General Trees: Prefix - Suffix – *7 points*)**

1. (a) **algorithm procedure** `ps_stat`
          local parameters
              `t_tree_tuples    T`
          global parameters
              `integer  c`
              `t_elts_vect    V`
          variables

   begin

   end algorithm procedure `ps_stat`
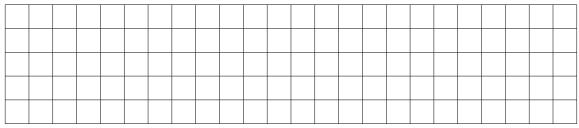
   (b) **algorithm function** `filling_stat` : integer
          local parameters
              `t_tree_tuples    T`
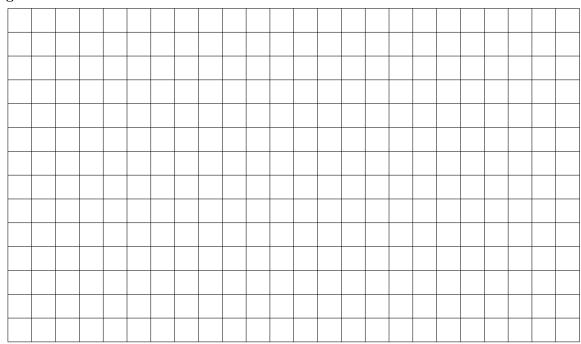   **global parameters**
              `t_elts_vect    V`

          variables

   begin

   end algorithm function `filling_stat`

2. **algorithm procedure** `ps_dyn`
    local parameters
        `t_dyn_tree T`
    global parameters
        `integer  c`
        `t_elts_vect    V`

    variables

    begin

    end algorithm procedure `ps_dyn`

Answers 3 (**B-trees: Insertions** − *2 + 6 points*)
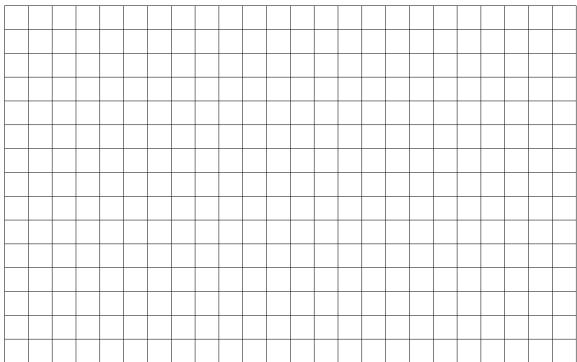
1. *Insertion of keys 18, 42 and 23:*

   After insertion of 18

   After insertion of 42

   After insertion of 23
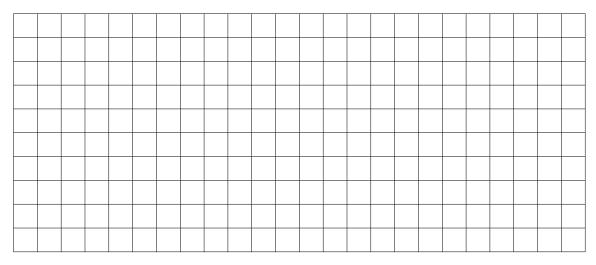
2. **Specifications:**

   The function `insert_rec` (x, B) inserts the key x in the tree B of type `t_Btree`, unless x is already in the tree. B is nonempty, and its root is not a full node (not a $2t$-node).

algorithm function `insert_rec` : `boolean`
    local parameters
        `t_element`   `x`
        `t_Btree`   `B`

    variables
        `integer`   `i, j`
begin
    `i ← search_pos(x, B)`

    if [_____] **then**

        if `B↑.children[1] <> NUL` **then**

        else

        end if
    else
        return `false`
    end if
end algorithm function `insert_rec`