

Prova com consulta. Duração: 90 minutos.

Nome do estudante: _____ N° _____

1. [2 valores] Quais os resultados das seguintes expressões em VDM?

a) $\{mk_{(1,2)}.#1, mk_{(2,2)}.#2\} <: \{1|->2, 2|->3, 3|->4\}$

b) $\{1|->2, 2|->3, 3|->3\} \text{ union } (\{4|->5\} ++ \{4|->6\})$

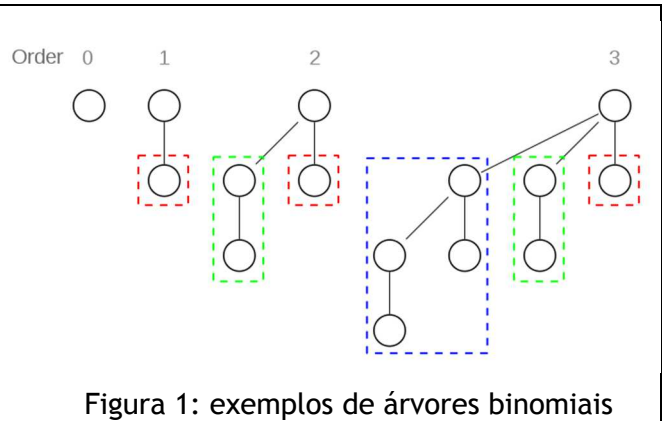
c) $\text{power} (\{1,2\} \text{ union elems } ([2,2] ^ ([2,3,4] ++ \{3|->5\})))$

d) $\text{dunion} \{\{1\}, \{2,3\}\} \text{ union dinter} \{\{1,3,4\}, \{2,3,4\}\}$

2. Considere a seguinte formalização em VDM++ para representar árvores binomiais (exemplos na figura a seguir).

```
class BinTree
types
  edge :: id: int
         children: set of int;
instance variables
  tree: set of edge;
  order: int;

-- inv1 "the root of the tree has id 0"
-- inv2 "a binomial tree of order k has 2k nodes"
-- inv3 "a binomial tree of order k has height k"
end BinTree
```



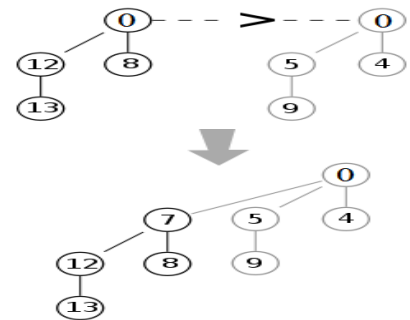
a) [2 valores] Assuma que existe uma função $root(b: BinTree)$ que retorna a raiz (do tipo $edge$) de uma árvore b . Formalize os invariantes $inv1$ (“a raiz da árvore tem id 0”) e $inv2$ (“Uma árvore binomial de ordem k tem 2^k nós”).

Prova com consulta. Duração: 90 minutos.

- b) [3 valores] Formalize o invariante **inv3** (“Uma árvore binomial de ordem k tem altura k ”). A altura de uma árvore é a distância máxima da raiz às folhas. Assuma que existe uma função $dist(f:edge, b:BinTree)$ que retorna a distância da folha f à raiz de uma árvore b .

Considere uma função *merge* que junta duas árvores de acordo com os seguintes requisitos:

- A árvore da esquerda é adicionada aos *edges* da árvore da direita
- A raiz da árvore da direita fica com *id* igual ao *id* mínimo menos um. Na figura ao lado, o nó correspondente à raiz da árvore da esquerda assumiu o valor 7 (isto é, $8-1$).



- c) [3 valores] Escreva a pré-condição da função *merge* que garanta que as árvores de entrada têm a mesma ordem e não têm nós repetidos (exceto a raiz com *id* 0)

- d) [3 valores] Escreva a função *merge* descrita anteriormente.

Prova com consulta. Duração: 90 minutos.

Nome do estudante: _____ Nº _____

3. [4 valores]

a. Determine (através do cálculo da *wp*) se o seguinte triplo de Hoare é verdadeiro

$\{x > -100 \wedge x \leq 100\}$ if $x < 0$ then $x := x + 100$ else skip; $y := 2 \cdot x$ $\{y > 0 \wedge y \leq 300\}$

b. Escreva uma pré-condição, *P*, que torne o seguinte triplo de Hoare verdadeiro:

```
int i, vec[];
{P}
while ( vec[i] <> x) {
    vec[i] := 0;
    i := i + 1;
}
{vec[j] = 0  $\wedge$  j >= 0  $\wedge$  j < i - 1  $\wedge$  vec[i] = x}
```

Prova com consulta. Duração: 90 minutos.

- c. Em que condições o ciclo anterior não termina? Qual é a função variante do ciclo? Justifique.

Boa sorte!