Lista 7

1. Tome como base a sequência de instruções abaixo (for i = 1 to 15 by 2 significa que i vai de 1 até 15 com um passo 2).

```
for i = 1 to 16
    MAKE-SET(x_i)
for i = 1 to 15 by 2
    UNION(x_i , x_{i+1})
for i = 1 to 13 by 4
    UNION(xi , x_{i+2})
UNION(x_1, x_5)
UNION(x_11, x_13)
UNION(x_1, x_10)
FIND-SET(x_2)
FIND-SET(x_9)
```

a) Desenhe a sequência final de conjuntos.

b) Considere uma implementação de lista encadeada com heurística de união ponderada. Como estarão essas listas no final do código?

c) Considere uma implementação por floresta com apenas a heurística de union by rank (sem path compression). Mostre a aparência dessas árvores ao fim da execução.

d) A partir da resposta da letra anterior, realize um find-set(x_2) com path compression e mostre a configuração final da árvore que representa seu conjunto.

2. Modifique o union da implementação por lista encadeada para entrelaçar as duas listas ao invés de concatenar. Note que dessa forma não é necessário manter um ponteiro para o fim da lista.

3. Utilize a representação por vetor da union find para encontrar o ancestral comum mais baixo entre dois nós de uma árvore binária (considerando que os nós da árvore binárias estão numerados de 0 até n). Note que isso se assemelha a uma utilização de uma tabela hash.

4. Construa uma função print-set nas três implementações, onde recebe um item x qualquer e imprime todo o conjunto ao qual ele pertence. Analise a complexidade obtida em cada implementação, onde n é o número de elementos do conjunto e m o número total de elementos.

5. Imagine que temos um vetor de ponteiros para itens, Item* vet[10] já alocados em seus devidos conjuntos por uma implementação de floresta, proponha uma forma de percorrer por todos os itens e contar quantos conjuntos disjuntos existem.

6. Considere o caso onde temos uma rede de computadores onde cada computador é denotado por um número. Sempre que é feita uma conexão entre uma máquina e outra anotamos em um arquivo a conexão da forma “x y” onde x é o número da primeira máquina e y é o número da segunda. Imagine então que a entrada para o seu programa será algo da forma:

5
1 3
2 4
2 1

Onde o primeiro número (5) é o número de máquinas da rede.
Proponha um algoritmo que identifique se a rede está completamente conectada (ou seja, consigo chegar de uma máquina a todas as outras pelas conexões feitas).