

Marco Ames

Disciplina: 4645G-04 - Algoritmos e Estruturas de Dados I

Trabalho 2

Etapa 1 - Ler Arquivo e criar a árvore.

O algoritmo percorre o arquivo lendo e gravando em variáveis, nome do pai, nome do filho e quantidade de terras do filho, adicionando a árvore em seguida utilizando o método *add()*.

Etapa 2 – Atualizar as terras dos filhos.

Usando um algoritmo de caminhamento em largura a quantidade de terras do nodo pai é dividida pela quantidade de filhos do mesmo. O valor é somado a quantidade de terras que cada filho já possuía, método *atualizaTerras()*.

Etapa 3 – Ver qual dos últimos filhos tem mais terras.

Primeiro verifica qual o nível mais alto que um nodo da lista ocupa, percorrendo em largura e gravando o nível mais alto já encontrado, método *maiorNivel()*.

Adicionando os demais nodos que estão no mesmo nível a uma lista, método *ultimoGuerreiroAux()*.

Finalmente se faz uma comparação entre os elementos da lista para descobrir qual nodo tem o maior número de terras como elemento, método *ultimoGuerreiro()*.

Um Código é gerado para visualizar a árvore em www.graphviz.org onde são mostrados o nome de cada guerreiro e sua quantidade de terras em cada nodo.

Casos de teste

04a.txt - Gorpanulaanix 7604

04b.txt - Retumiattox 9413

05c.txt - Cridenaarnax 6759

06b.txt - Frinvalrix 9081

07a.txt - Nepcriclox 8480

08a.txt - Flermonrattox 10772

09b.txt - Docmonmonflex 8513

10c.txt - Furganrix 8673