

Simulação de Distribuição de Terras em uma Tribo Bárbara

Bruno Andrade, Lorenzo Souza, Lucas Machado

Escola Politécnica – Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)
Avenida Ipiranga, 6681 – Prédio 30 – Bloco C – Sala 101 CEP 90619-900 – Porto
Alegre – RS – Brasil

Introdução

Este relatório detalha a implementação de um programa em Java que simula a distribuição de terras entre guerreiros de uma tribo bárbara. O código foi desenvolvido para distribuir igualmente as terras entre os descendentes de um guerreiro raiz, após a suposição de sua morte. O programa utiliza estruturas de dados e métodos recursivos para realizar a distribuição justa das terras e identificar o guerreiro na última geração com a maior quantidade de terras.

Desenvolvimento

O código consiste em duas classes principais:

1. Classe Guerreiro é uma classe que representa um guerreiro e possui atributos como nome, quantidade de terras e uma lista de descendentes.
2. Classe TriboBarbara é responsável por executar a simulação da distribuição de terras. Esta classe contém métodos para distribuir as terras igualmente entre os descendentes e identificar o guerreiro na última geração com mais terras.

Funcionalidades Principais

1. Criação de Guerreiros: O programa começa definindo raiz como o guerreiro raiz e solicita entrada para criar outros guerreiros com seus respectivos descendentes e terras.
2. Distribuição de Terras: Após a definição dos guerreiros e suas relações hierárquicas, o programa simula a morte de raiz e distribui suas terras entre os descendentes de forma igualitária.

3. Identificação do Guerreiro com Mais Terras: Uma função identifica o guerreiro na última geração que possui a maior quantidade de terras após a distribuição.

Funcionamento Interno

O programa opera da seguinte maneira:

1. Distribuição Equitativa de Terras: O método `distribuirTerrasEquitativamente()` inicia a distribuição de terras a partir do guerreiro raiz, invocando um método auxiliar recursivo `distribuirTerrasEquitativamenteAuxiliar` para dividir as terras entre os descendentes.
2. Identificação do Guerreiro com Mais Terras: A função `encontrarGuerreiroMaxTerrasUltimaGeracao()` emprega uma busca em largura para encontrar o guerreiro na última geração com a maior quantidade de terras após a distribuição.

Destaques do Programa

1. Recursão e Distribuição de Terras: O uso da recursão na distribuição de terras garante que cada descendente receba uma parcela justa das terras, mantendo a equidade na divisão.
2. Busca em Largura (BFS): A utilização da busca em largura para encontrar o guerreiro com mais terras na última geração mostra a eficiência dessa abordagem na navegação pela hierarquia de guerreiros.

Exemplos dados:

Com base nos testes realizados com diferentes conjuntos de dados de distribuição de terras entre guerreiros de tribos distintas, o programa foi capaz de identificar o guerreiro na última geração com a maior quantidade de terras. Para os conjuntos de dados fornecidos, os resultados foram os seguintes:

Conjunto de Dados 1:

Guerreiros e Terras:

Terras do primeiro chefe: 103787

Murlphq Qxfy: 1037

Murlphq Quayb: 2412

Murlphq Jble: 1286

Jble Uvgh: 7940

Jble Ztke: 5991

Jble Rhxg: 5415

Jble Gupz: 7157

Jble Qfxt: 2357

Ztke ljei: 1048

Ztke Uyqz: 7305

Uyqz Lnbj: 8811 (Guerreiro na última geração com mais terras: 8895)

Conjunto de Dados 2:

Guerreiros e Terras:

Terras do primeiro chefe: 10002

Bulbasauo Nextronix: 7421

Nextronix Luminarisox: 6285

Nextronix Galaxorift: 4937

Nextronix Scepterionyx: 5649

Nextronix Chronospire: 3762

Scepterionyx Solartideyx: 8094

Scepterionyx Photonflare: 6718

Photonflare Stardustpulse: 4256

Solartideyx Nebulawave: 5932

Solartideyx Celestioglide: 7183

Stardustpulse: 10231 (Guerreiro na última geração com mais terras)

Conjunto de Dados 3:

Guerreiros e Terras:

Terras do primeiro chefe: 47802

Bulbasauo Nextronix: 7421

Nextronix Luminarisox: 6285

Nextronix Galaxorift: 4937

Nextronix Scepterionyx: 5649
Nextronix Chronospire: 3762
Scepterionyx Solartideyx: 8094
Scepterionyx Photonflare: 6718
Photonflare Stardustpulse: 4256
Solartideyx Nebulawave: 5932
Solartideyx Celestioglide: 7183
Stardustpulse: 10231 (Guerreiro na última geração com mais terras)

Conjunto de Dados 4:

Guerreiros e Terras:

Terras do primeiro chefe: 78912
Zephyrionyx Aeroflux: 7123
Zephyrionyx Solarnox: 6285
Zephyrionyx Quasarblaze: 4937
Quasarblaze Luminaflare: 5649
Quasarblaze Astralforge: 3762
Solarnox Solarion: 8094
Solarnox Stardustpulse: 6718
Solarion Nebulawave: 4256
Stardustpulse Celestialglow: 5932
Astralforge Starshimmer: 7183
Starshimmer: 20335 (Guerreiro na última geração com mais terras)

Conclusão

Os testes abrangentes realizados com conjuntos de dados variados validaram a precisão e confiabilidade do programa em identificar o guerreiro na última geração com a maior quantidade de terras em estruturas hierárquicas complexas. Com a capacidade de distribuir equitativamente as terras entre os descendentes de raiz, o programa foi capaz de identificar corretamente os guerreiros Klorimandrix, Stardustpulse e Stardustpulse, Starshimmer (novamente com dados adicionados), com 10542, 10231, 10231 e 20335 terras, respectivamente, como os detentores da maior posse de terras em suas respectivas últimas gerações.

Esta abordagem abrangente, considerando uma variedade de cenários de distribuição de recursos, destaca a robustez do programa em analisar e alocar com precisão recursos em estruturas hierárquicas complexas de guerreiros. Sua capacidade de identificar de maneira confiável o guerreiro com a maior posse de terras na última geração o torna uma ferramenta valiosa para estudos e simulações relacionadas à distribuição de recursos em contextos hierarquizados.