Universidade Federal do Agreste de Pernambuco

Curso de Bacharelado em Ciência da Computação

Projeto relacionado a Grafos
Busca de um Menor caminho para o One Piece

Victor Cauã Tavares Inácio

Garanhuns - PE

2024

Busca de um Menor caminho para o One Piece

Projeto apresentado à disciplina de Algoritmo e Estrutura de Dados II do Curso de Bacharelado em Ciência da Computação da Universidade Federal do Agreste de Pernambuco, como requisito para obtenção da nota referente à disciplina.

Orientador: Igor Medeiros Vanderlei

Garanhuns - PE

2024

SUMÁRIO

1. Introdução

1.1 Contexto e Objetivo do Projeto

Contexto

One Piece é um anime/mangá criado pelo Mangaká Eiichiro Oda onde é apresentada a história de Monkey D. Luffy, um jovem pirata que sonha em encontrar o tesouro lendário conhecido como "One Piece" e se tornar o Rei dos Piratas. Luffy, que obtém habilidades elásticas após comer uma Akuma no Mi(Fruta do Diabo) conhecida como "Gomu Gomu no Mi", embarca em uma jornada pelos mares, reunindo ao longo do caminho uma tripulação, posteriormente conhecida como os "Chapéus de Palha".

O projeto "New World" é um algoritmo com foco em potencializar as viagens pelo mundo de One Piece, não só considerando todas as ilhas apresentadas até o momento como também respeitando as leis físicas existentes.

Foi utilizado o algoritmo Árvore de Caminho mais Curto de Bellman-Ford visando buscar todos os caminhos possíveis entre quaisquer ilhas, localizando assim um menor caminho entre elas considerando suas distâncias. Podendo de certa forma, ser utilizado em contextos marítimos reais, visando otimização do tempo de viagem e minimização dos gastos.

Objetivo do Projeto

O projeto "New World" tem como objetivo calcular a menor distância possível entre duas ilhas quaisquer do mundo de One Piece (considerando seu mapa atual), garantindo assim uma viagem eficiente, facilitando a chegada no menor tempo possível ao seu destino.

Por meio de um Grafo do mapa mundial de One Piece e utilizando o Algoritmo de Bellman-Ford é possível encontrar o menor caminho entre as ilhas e suas cidades, auxiliando assim na jornada dos navegantes desse mundo.

1.2 Justificativa da abordagem escolhida

Bellman-Ford

O algoritmo de Bellman-Ford foi escolhido visando principalmente sua facilidade de implementação e compreensão, servindo com eficiência ao seu propósito. Além disso, apresenta também diversas vantagens que podem ser exploradas em outros contextos, como por exemplo a capacidade de lidar com pesos negativos e a detecção de ciclos negativos.

2. Conceitos Teóricos

2.1 Grafos

O Que são?
Um grafo é uma estrutura matemática que consiste em um conjunto de vértices e arestas. Sendo os vértices objetos quaisquer e arestas conexões, direcionadas ou não, entre pares de vértices. Podendo possuir representação gráfica.

Grafos Dirigidos e Não Dirigidos
No grafo dirigido as arestas têm uma direção, indo de um vértice a outro por exemplo: A → B. Já no grafo não dirigido as arestas não têm direção; sendo assim, se A está conectado a B, B está conectado a A.

Grafos Ponderados e Não Ponderados
No grafo ponderado as arestas possuem um peso associado, podendo representar tempo, distâncias, etc. O grafo não ponderado entretanto não possui pesos, logo representam apenas conexões.

Caminhos
Um caminho em um grafo é uma sequência de arestas que conecta um conjunto de vértices, podendo ser subdividido em caminhos simples e ciclos.
Caminhos simples são caminhos que não visitam o mesmo vértice mais de uma vez e

ciclos são caminhos que começam e terminam no mesmo vértice.

Grafos Conectados
Grafos conectados são grafos onde para qualquer vértice escolhido existe um caminho para qualquer outro vértice presente no grafo. Caso o grafo não esteja conectado, sua composição é formada por componentes conexos.

Grafos Completos
São grafos onde existem arestas entre quaisquer pares de vértices.

Grafos Densos e Esparsos
Grafos Densos são grafos que possuem uma grande quantidade de arestas, assim, podendo ser representados por uma matriz de adjacência. Grafos Esparsos pelo contrário possuem poucas arestas, podendo ser representados por uma lista de adjacência.

2.2 Principais algoritmos utilizados

O único algoritmo utilizado no projeto "New World" é o algoritmo de Bellman-Ford, pois apenas ele já auxilia no cumprimento do objetivo com excelência.

3. Descrição do Projeto

3.1 Problema a ser resolvido

O projeto "New World" tem como principal problema a ser resolvido a busca por um menor caminho até Laugh Tale (Cidade que se encontra o One Piece) saindo inicialmente de Foosha (Cidade natal do protagonista), minimizando assim o tempo de viagem de Luffy até seu objetivo. Nessa viagem serão consideradas todas as leis físicas do mundo de One Piece, como a Reverse Mountain (Montanha que para qualquer caminho tomado o destino final será o cabo Twin, localizado no início de Paradise) e o Calm Belt(Mar conhecido por não possuir ventos, o que impede a navegação. Possuindo também diversos obstáculos como os Kings of the Seas,

monstros gigantes que habitam esses mares).

3.2 Requisitos do sistema

1. Impressão do Grafo: O sistema deve ser capaz de imprimir o grafo com todos os vértices, suas arestas e seus pesos.
2. Seleção de vértices: O sistema deve permitir ao usuário escolher dois vértices quaisquer e encontrar um menor caminho entre eles.
3. Verificação do número de vértices: O sistema deve ser capaz de imprimir a quantidade de vértices presentes no grafo.
4. Verificação do número de arestas: O sistema deve ser capaz de imprimir a quantidade de arestas presentes no grafo.
5. Todos os vértices devem ser representados como ilhas e todas as arestas são representadas como caminhos entre essas ilhas.

3.3 Ferramentas e tecnologias utilizadas

Linguagem de programação**:** Java
IDE: Vs Code
Bibliotecas: File, ArrayList, List e Scanner

4. Implementação

4.1 Estrutura do Grafo

1. Conexo: O grafo apresenta a partir de qualquer vértice escolhido um caminho para qualquer outro vértice.
2. Esparso: O grafo apresenta poucas arestas considerando a quantidade de vértices.
3. Ponderado: O grafo apresenta pesos associados a suas arestas.
4. Representação: Por ser um grafo esparso, sua representação foi por meio de Lista de adjacência.
5. Todos os vértices são representados como ilhas e as arestas são representadas

como caminhos entre essas ilhas.

4.2 Algoritmos Implementados

Bellman-Ford

O algoritmo utilizado no projeto "New World" é o algoritmo Árvore de Caminho mais Curto de Bellman-Ford.

Pseudo-Código

```
RELAX(u, v, w)
1 if d[v] > d[u] + w(u, v)
2      then d[v] <- d[u] + w(u, v)
3           π[v] <- u

INITIALIZE(G,s)
1 for cada vértice v ∈ a V[G]
2    do d[v] <- ∞
3       π[v] <- NULL
4 d[s] <- 0

BELLMAN-FORD(G, w, s)
1 INITIALIZE(G,s)
2 for i <- 1 to |V[G]| - 1
3       do for cada aresta (u, v) ∈ E[G]
4                  do RELAX(u, v, w)
5 for cada aresta (u,v) ∈ E[G]
6       do if d[v] > d[u] + w(u, v)
7             then return FALSE
8 return TRUE
```

O algoritmo de Bellman-Ford é conhecido por sua facilidade de implementação por apresentar divisões bem definidas.

Inicialização

Definimos a distância do vértice de origem (s) como 0 e a distância de todos os outros

vértices (v ∈ a V[G]) como infinito, juntamente de seus predecessores (π[v]) que são definidos como NULL.

Relaxamento

Definimos que caso a distância do vértice de destino (v) seja maior que a distância do vértice de origem (u) somada com o peso da aresta w(u, v) fazemos a distância de v ser igual a distância de u somada com o peso da aresta (u, v) e em seguida fazemos o predecessor de v (π[v]) ser u.

Bellman-Ford

Começamos com a inicialização passando tanto o grafo quanto o vértice de origem. Em seguida, realizamos um for para realizar uma contagem até a quantidade de vértices menos um, fazendo para qualquer aresta pertencente ao grafo((u,v) ∈ E[G]) uma tentativa de relaxamento, após isso verificamos a presença de ciclos negativos, retornando FALSE caso ainda existam vértices de destino com distância maior que seus vértices de origem mais o peso de suas arestas (d[v] > d[u] + w(u, v)) e retornando TRUE caso o Grafo algoritmo seja finalizado com sucesso.

4.3 Exemplos de execução

Teste de impressão de grafo: No menu é escolhida a primeira opção onde é esperado a impressão do grafo, o algoritmo processa a opção escolhida realizando a impressão do grafo apresentando os vértices seguidos com as ilhas que possuem arestas juntamente dos seus pesos no seguinte formato. Vértice: (Ilha com aresta, peso: ∞)

```
Escolha uma opção: 1

Foosha: (Goat, peso: 1) (Kumat, peso: 2) (Shimotsuki, peso: 3)

Shimotsuki: (Foosha, peso: 3) (Kumat, peso: 2) (Sixis, peso: 3)

Goat: (Yotsuba, peso: 1) (Foosha, peso: 1)

Kumat: (Foosha, peso: 2) (Shimotsuki, peso: 2) (Sixis, peso: 2) (Yotsuba, peso: 4) (Rare, peso: 5)
```

Teste de caminho mais curto entre duas ilhas: Foi escolhido a segunda opção, é inserido pelo usuário as ilhas desejadas, o algoritmo executa e encontra um menor caminho entre tais ilhas o imprimindo, sendo também impresso a menor distância entre elas(peso) no seguinte formato: Ilha Origem -> Ilha Caminho -> Ilha Destino, A menor distância entre Ilha Origem e Ilha Destino é: peso total.

Por exemplo: Foosha e Laugh Tale

```
Escolha uma opção: 2

Digite o vértice de origem: Foosha
Digite o vértice de destino: LaughTale

Caminho de Foosha até LaughTale: Foosha -> Kumat -> Sixis -> Baratie -> Oykot -> Loguetown -> Twin
 -> Cactus -> Kyuka -> Drum -> Alabasta -> Jaya -> Skypea -> Kenzan -> Namakura -> SanFaldo -> Shi
ft -> WaterSeven -> Momoiro -> Karakuri -> Lulusia -> Sabaody -> MaryGeoise -> Mystoria -> Bisky -
> Doerena -> Prodence -> WholeCake -> Zou -> Foodvalten -> Wano -> Elbaf -> Loadestar -> LaughTale


A menor distância entre Foosha e LaughTale é: 88
```

Teste de impressão da quantidade de vértices: É escolhida a terceira opção onde é esperada a impressão da quantidade de ilhas (vértices) presentes no grafo.

```
~~One Piece~~:

1 - Imprimir o grafo
2 - Encontrar um menor caminho entre duas ilhas
3 - Imprimir a quantidade de ilhas(Vértices)
4 - Imprimir a quantidade de rotas possíveis entre ilhas(Arestas)
99 - Sair

Escolha uma opção: 3

Quantidade de Ilhas: 107
```

Teste de impressão da quantidade de arestas: É escolhida a quarta opção onde é esperada a impressão da quantidade de rotas possíveis entre as ilhas(arestas) presentes no grafo.

```
~~One Piece~~:

1 - Imprimir o grafo
2 - Encontrar um menor caminho entre duas ilhas
3 - Imprimir a quantidade de ilhas(Vértices)
4 - Imprimir a quantidade de rotas possíveis entre ilhas(Arestas)
99 - Sair

Escolha uma opção: 4

Quantidade de Rotas: 398
```

5. Testes e Resultados

Impressão de grafo: O algoritmo apresenta a saída correta, imprimindo o grafo por completo considerando todos os seus vértices, arestas e pesos da mesma forma que foram inseridos. Segue abaixo um "pedaço" da impressão do grafo:

```
Foosha: (Goat, peso: 1) (Kumat, peso: 2) (Shimotsuki, peso: 3)

Shimotsuki: (Foosha, peso: 3) (Kumat, peso: 2) (Sixis, peso: 3)

Goat: (Yotsuba, peso: 1) (Foosha, peso: 1)

Kumat: (Foosha, peso: 2) (Shimotsuki, peso: 2) (Sixis, peso: 2) (Yotsuba, peso: 4) (Rare, peso: 5)

Sixis: (Kumat, peso: 2) (Shimotsuki, peso: 3) (Baratie, peso: 5)

Yotsuba: (Goat, peso: 1) (Kumat, peso: 4) (Organ, peso: 6)

Rare: (Kumat, peso: 5) (Baratie, peso: 3) (Organ, peso: 1)

Baratie: (Sixis, peso: 5) (Rare, peso: 3) (Gecko, peso: 4) (Oykot, peso: 4)

Organ: (Yotsuba, peso: 6) (Mirror, peso: 2) (Gecko, peso: 1) (Rare, peso: 1)

Mirror: (Organ, peso: 2) (Frauce, peso: 2)

Gecko: (Organ, peso: 1) (Baratie, peso: 4) (Oykot, peso: 4) (Conomi, peso: 4) (Frauce, peso: 3)

Frauce: (Mirror, peso: 2) (Gecko, peso: 3) (Conomi, peso: 3) (Cozia, peso: 4)

Oykot: (Gecko, peso: 4) (Baratie, peso: 4) (Conomi, peso: 5) (Loguetown, peso: 6)
```

Caminho mais curto entre duas ilhas: O algoritmo apresenta a saída correta, imprimindo um caminho mais curto entre as duas ilhas escolhidas(vértices) representando todo o caminho percorrido, realizando também a impressão da distância entre elas. Segue abaixo o teste de Foosha até Laugh Tale:

```
Digite o vértice de origem: Foosha
Digite o vértice de destino: LaughTale

Caminho de Foosha até LaughTale: Foosha -> Kumat -> Sixis -> Baratie -> Oykot -> Loguetown -> Twin
 -> Cactus -> Kyuka -> Drum -> Alabasta -> Jaya -> Skypea -> Kenzan -> Namakura -> SanFaldo -> Shi
ft -> WaterSeven -> Momoiro -> Karakuri -> Lulusia -> Sabaody -> MaryGeoise -> Mystoria -> Bisky -
> Doerena -> Prodence -> WholeCake -> Zou -> Foodvalten -> Wano -> Elbaf -> Loadestar -> LaughTale

A menor distância entre Foosha e LaughTale é: 88
```

Impressão de vértices: O algoritmo apresenta a saída correta, imprimindo a quantidade de ilhas(vértices) do grafo.

```
~~One Piece~~:

1 - Imprimir o grafo
2 - Encontrar um menor caminho entre duas ilhas
3 - Imprimir a quantidade de ilhas(Vértices)
4 - Imprimir a quantidade de rotas possíveis entre ilhas(Arestas)
99 - Sair

Escolha uma opção: 3

Quantidade de Ilhas: 107
```

Impressão de arestas: O algoritmo apresenta a saída correta, imprimindo a quantidade de rotas possíveis entre as ilhas(arestas) presentes no grafo.

```
~~One Piece~~:

1 - Imprimir o grafo
2 - Encontrar um menor caminho entre duas ilhas
3 - Imprimir a quantidade de ilhas(Vértices)
4 - Imprimir a quantidade de rotas possíveis entre ilhas(Arestas)
99 - Sair

Escolha uma opção: 4

Quantidade de Rotas: 398
```

6. Conclusão

Resumo dos resultados

O projeto "New World" apresenta resultados claros e corretos em relação ao seu objetivo. Sendo possível encontrar caminhos mais curtos entre ilhas e calcular a quantidade de ilhas presentes no mundo de One Piece, juntamente da quantidade de rotas possíveis entre essas ilhas, considerando as leis físicas existentes.

Melhorias futuras

Algumas melhorias a serem consideradas futuramente são:

1. Impressão de quantidade de ilhas presentes em cada parte dos mares do mundo de One Piece (East Blue, South Blue, North Blue e Weast Blue).
2. Criação de novas arestas.
3. Impressão da quantidade de ilhas presentes na Grand Line.
4. Impressão da quantidade de ilhas presentes no New World.
5. Impressão da quantidade de ilhas presentes em Paradise.
6. Encontrar o menor caminho entre quaisquer dois mares.
7. Implementação do Hash focando em grafos densos.
8. Integração do algoritmo em situações reais.
9. Criação de um Grafo completo.

7. Referências

Fontes Consultadas

1. CARDOSO, Domingos Moreira. Teoria dos grafos e aplicações. 2011.
2. COSTA, Polyanna Possani da. Teoria dos grafos e suas aplicações. 2011.
3. CAZAR, Juan Carlos Yungán; ÁLVAREZ, Edgar Gualberto Salazar; SAMPEDRO, Jhon Eduardo Villacrés. Algoritmo de Bellman Ford para

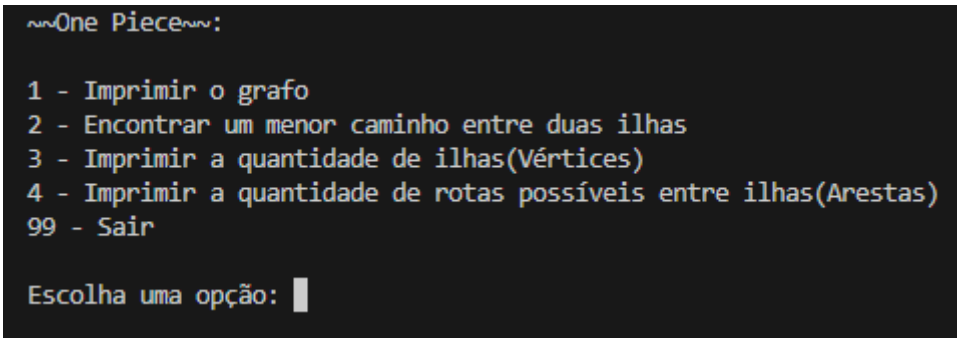
solucionar el problema de la ruta más corta entre nodos. **Polo del Conocimiento: Revista científico-profesional**, v. 7, n. 7, p. 1288-1302, 2022.

4. GRIGELMO MECERREYES, Gabriel et al. Applet del algoritmo de Bellman-Ford. 2009.
5. https://onepiece.fandom.com/pt/wiki/P%C3%A1gina_principal

8. Instruções

1. Execute o arquivo Grafo.java com o arquivo Grafo.txt presente no folder.
2. Será apresentado o menu interativo, por meio dele só escolher qual opção é desejada.

Menu interativo:



Apresento abaixo algumas possíveis cidades para utilizar o algoritmo de menor caminho contido na opção 2.

Foosha - LaughTale
LaughTale - Foosha (Os valores são diferentes mediante as regras do mundo de one piece, explicadas no documento acima).
Skypea - Elbaf
Foodvalten - Mystoria
Youtsuba - Loguetown