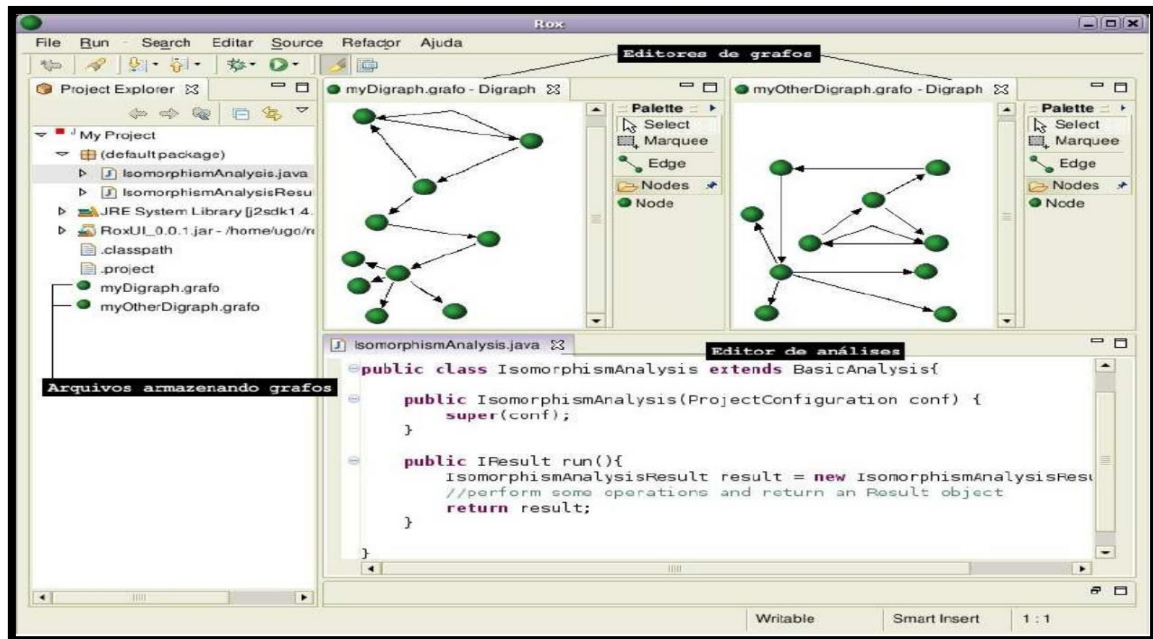


Alunos: Leonardo Fronza, Rodrigo Fernandes

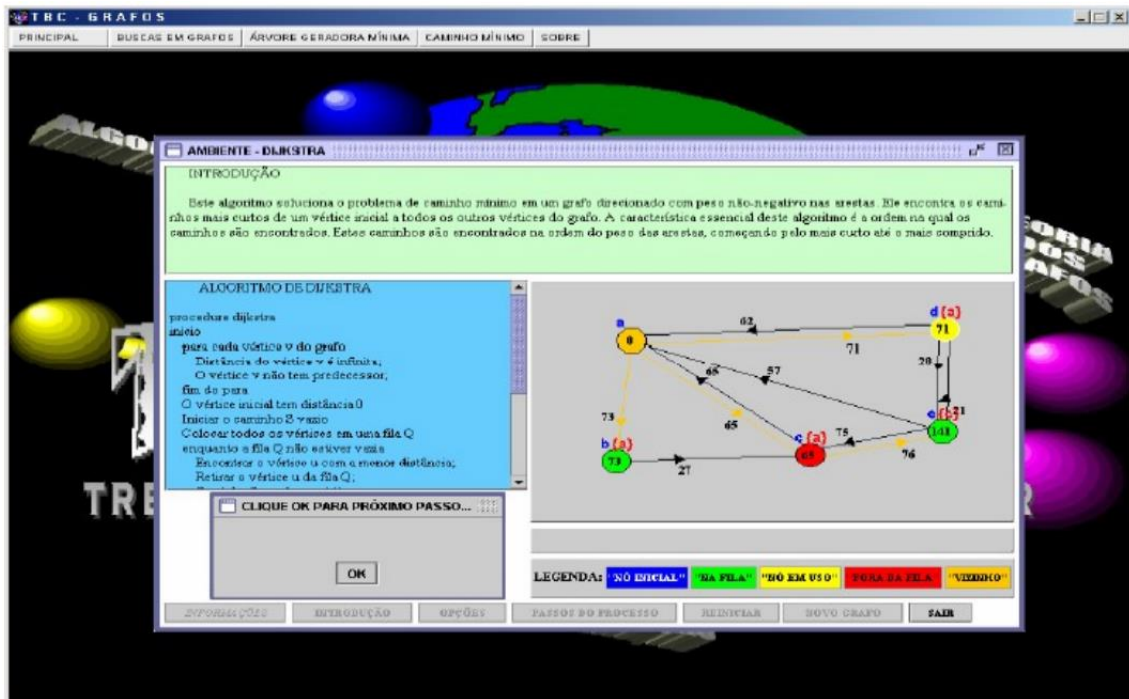
Questão 1:

Rox Graph Theory

É um framework open source destinado a construção de aplicações baseadas em grafos. O projeto é dividido em duas partes principais, a primeira parte é a API para Grafos com o objetivo de prover aos usuários a possibilidade de lidar com grafos sem necessariamente fazer uso de uma interface gráfica, conforme imagem abaixo:

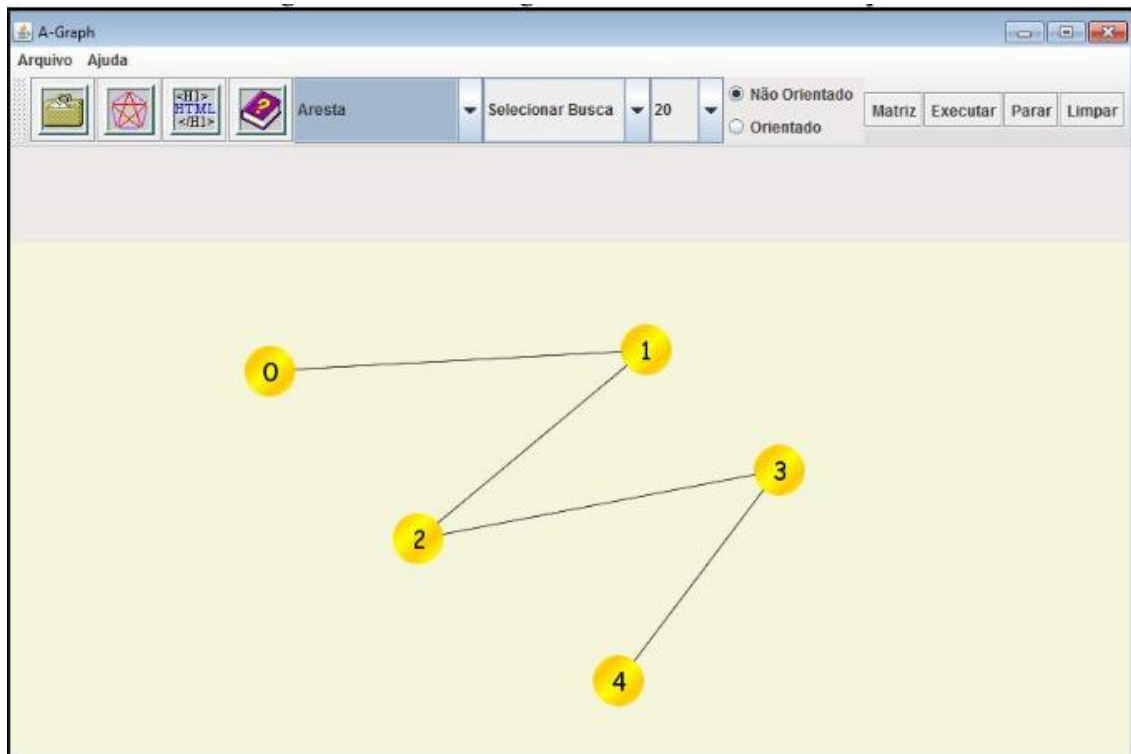


A segunda parte é a interface gráfica baseada no Java Eclipse, que permite que o Software tenha um editor visual para grafos e possa executar algoritmos baseados nas estruturas criadas, conforme imagem abaixo:



A-Graph

Implementado na linguagem Java, a qual utiliza os recursos de computação gráfica bidimensional suportada pela API Java2D. A aplicação roda em modo stand-alone, o que significa que é necessário somente o executável para que a mesma funcione, ficando assim sem dependências de terceiros. O aplicativo A-Graph oferece uma interface de interação com a qual os estudantes podem desenhar grafos, com ou sem direção, definindo livremente a sua geometria e consequentemente a topologia subjacente. Permite a execução dos algoritmos de busca em grafos, de maneira didática, inclusive implementando a heurística de escolha livre do vértice inicial a partir do qual a busca se iniciará. A interface gráfica da ferramenta permite o desenho livre dos grafos, conforme imagem abaixo:



Ela foi elaborada com o propósito de ser utilizada como suporte ao ensino-aprendizado. A qual possui foco na execução dos algoritmos de busca em largura (BFS) e busca em profundidade (DFS), assim como também exibe as matrizes de incidência e de adjacência. Na imagem abaixo podemos observar a funcionalidade de exibição das matrizes de incidência e de adjacência.

Matriz						Matriz				
Matriz de Adjacência						Matriz de Incidência				
	v0	v1	v2	v3	v4		0↔1	3↔4	2↔3	1↔2
v0	0	1	0	0	0	v0	1	0	0	0
v1	1	0	1	0	0	v1	1	0	0	1
v2	0	1	0	1	0	v2	0	0	1	1
v3	0	0	1	0	1	v3	0	1	1	0
v4	0	0	0	1	0	v4	0	1	0	0

Ela também permite a substituição dos vértices por imagens diversas, tais como roteadores, onde é possível a modelagem de problemas em outras áreas, como a área de redes. Sendo assim, a ferramenta atinge o seu propósito de executar os algoritmos BFS e DFS, permitir a criação dos grafos e dígrafos de forma livre e apresentar as matrizes de incidência e de adjacência, tornando-se válida para auxiliar no ensino-aprendizagem.

Questão 2:

Neo4j

O Neo4j é um banco de dados de gráficos nativo altamente escalonável, desenvolvido especificamente para alavancar não apenas dados, mas também relacionamentos de dados. Ele suporta aplicativos transacionais e análises de gráficos, ajudando os cientistas de dados a obter novas perspectivas sobre os dados. Sua integração de dados acelera a destilação de dados tabulares e big data em gráficos. Sua linguagem de consulta de gráficos é o Cypher, a qual é a ponte para as ferramentas analíticas de big data, auxiliando na visualização do grafos e com isto melhorando os benefícios da tecnologia gráfica em toda a organização, além também de possuir sua arquitetura corporativa e suportando dados gráficos massivos e com isto tornando suas consultar mais simples e diretas.

Ela é feito de Nós, Labels, Relacionamentos e propriedades;

- a) Nós são os registros
- b) Labels são conjunto agrupado de nós
- c) Relacionamentos são as conexões entre os nós
- d) Propriedades são definições de um nós (Direção, Agrupamento ect)

Suas aplicações e usos são:

- Fraud Detection
- Real-time recommendation
- Social network
- Identity and access management
- Network ant it operations

Suporta nas seguintes plataformas:

- Windows
- Linux/ Unix
- Mac osx

Como utiliza cypher, sua linguagem é declarativa, de inspiração SQL para descrever padrões em gráficos utilizando uma sintaxe ascii-art. Ela também nos permite selecionar, inserir ou excluir dados do gráfico sem a necessidade de descrever exatamente como fazê-lo.

Sua aplicação integra nativamente com: Java, C#, Python, Javascript, PHP, Go e Restful API.

OrientDB

O OrientDB é o banco de dados de gráficos mais rápido do mundo assim pode-se dizer. Pois um estudo realizado pela IBM e do Instituto de Tecnologia de Tóquio mostrou que o OrientDB é 10 vezes mais rápido que o Neo4j em operações gráficas entre todas as

cargas de trabalho. A qual gere vantagem competitiva e acelere a inovação com novos fluxos de receita. Ele também é o primeiro sistema de gerenciamento de banco de dados NoSQL multi-modelo de código aberto que combina o poder dos gráficos com modelos de documentos, chave / valor, reativos, orientados a objetos e geoespaciais em um único banco de dados operacional escalável e de alto desempenho.

O OrientDB ganhou espaço devido a era do big data, a qual as empresas precisam de bancos de dados que suportam mais do que apenas um único modelo de dados. Caso contrário, é impossível eles desbloquearem o verdadeiro poder de seus dados de maneira eficiente.

Seus principais aspectos são:

- a) Banco de dados multi-modelo
- b) Armazena até 120.000 registros por segundo
- c) Relacionamentos são links físicos para os registros, não há necessidade de junções
- d) Garante melhor uso da memória RAM
- e) Percorre partes ou árvores inteiras e gráficos de registros em milissegundos
- f) A velocidade não é afetada pelo tamanho do banco de dados; grandes conjuntos de dados são facilmente acomodados

Ele também fornece Backups incrementais, segurança incomparável, suporte 24x7, perfil de consulta, configuração de cluster distribuído, gravação de métricas e monitoramento ao vivo com alertas configuráveis

Dex/Sparksee

O Sparksee (anteriormente conhecido como DEX) torna o espaço e o desempenho compatíveis com uma pequena área ocupada e uma análise rápida de grandes redes. Ele está disponível nativamente para .Net, C ++, Python, Objective-C e Java, e abrange todo o espectro de sistemas operacionais. Ele também é o primeiro banco de dados de gráficos disponível para iOS e Android.

Seus principais aspectos são:

- a) Espaço reduzido de memória fácil de gerenciar;
- b) O gráfico é representado através de estruturas de dados de bitmap que permitem altas taxas de compactação;
- c) As adjacências do nó são representadas por bitmaps para minimizar sua presença;
- d) Cada valor no banco de dados é representado apenas uma vez, evitando replicação desnecessária.

Possui grande eficiência de E / S recuperando dados do banco de dados, onde:

a) Cada um dos bitmaps é particionado em pedaços que cabem nas páginas do disco para melhorar a localidade de E / S.

b) O número de vezes que cada página de dados é trazida para a memória é minimizado com políticas avançadas de E / S

c) Ele fornece acesso direto aos buffers do SO, evitando a decodificação e a codificação de outras soluções.

Ele também fornece o aumentando o desempenho no máximo da aplicação, onde é:

a) Usando bitmaps, as operações são calculadas com instruções lógicas binárias que simplificam a execução em processadores em pipeline;

b) O núcleo do C ++ evita a execução de sobrecarga e o gerenciamento complexo de memória, em oposição aos mecanismos baseados em Java;

c) Os tempos de resposta de consulta de alta latência e baixa latência garantem alto desempenho, mesmo sob condições de estresse;

d) Ampla compatibilidade para orientação do usuário;

e) A indexação nativa completa permite um acesso extremamente rápido a cada uma das estruturas de dados do gráfico;

f) API de baixo nível com acesso direto às funcionalidades do mecanismo principal.

HyperGraphDb

O HyperGraphDB é um mecanismo de armazenamento de dados de código aberto de propósito geral, baseado em um poderoso formalismo de gerenciamento de conhecimento conhecido como hiper gráficos direcionados. Embora seja um modelo de memória persistente projetado principalmente para gerenciamento de conhecimento, IA e projetos da Web semântica, também pode ser usado como um banco de dados orientado a objetos incorporado para projetos Java de todos os tamanhos.

Possui os seguintes recursos:

a) Poderosa modelagem de dados e representação de conhecimento.

b) Armazenamento orientado a gráficos.

c) N-ária, relações de ordem superior (arestas) entre nós do gráfico.

d) Travessias de gráfico e consultas em estilo relacional.

e) Indexação personalizável.

f) Gerenciamento de armazenamento personalizável.

g) Esquema de banco de dados dinâmico extensível por meio de digitação personalizada.

h) Banco de dados Java OO pronto para uso.

i) Totalmente transacional e multithread, MVCC / STM. Gravações e leituras simultâneas sem bloqueio!

j) Estrutura P2P para distribuição de dados.

k) Componentes da aplicação

O aplicativo HyperGraphDB implementam vários modelos de domínio, padrões, algoritmos e ferramentas específicas do domínio, aproveitando sua generalidade. Cada entidade nesses componentes é, em última análise, um átomo do HyperGraphDB, o que torna possível integrá-los e compor esses componentes naturalmente, que são: JSON, WordNet, TopicMaps, RDF via Sail, OWL 2.0 e Protege Plugin

GRAPHOBJ

É uma biblioteca de programação para grafos implementada em duas versões, uma em linguagem Java e outra em Linguagem C++. A principal finalidade deste framework é ser didática. Ela possui uma estrutura de classes que permite a criação de grafos e dígrafos, além de permitir a associação de custo às arestas.

Também utiliza-se das estruturas de dados disponibilizadas pela STL para organizar a estrutura interna do grafo, além de trazer consigo um conjunto de algoritmos voltados à aplicação em grafos. Este framework permite a utilização de seus métodos pelo programador para o desenvolvimento de uma ferramenta que crie grafos e aplique algoritmos sobre os mesmos. No entanto, por estar ainda em desenvolvimento, qualquer funcionalidade não encontrada na biblioteca que se faça necessária ao programador deve ser implementada pelo mesmo.

Esta biblioteca possui uma classe abstrata chamada BaseGraph que possui atributos e métodos comuns às classes Graph e Digraph, as quais permitem ao usuário a criação de grafos e dígrafos respectivamente. Assim como no caso do grafo, a classe Edge para criação de arestas também é abstrata, com características comuns às classes de arestas dirigidas, DirectedEdge, e não dirigidas, UndirectedEdge. Além disso, pode-se perceber que há uma relação de agregação, onde um grafo deve agregar pelo menos um vértice, e uma associação entre vértice e aresta, na qual um vértice não precisa necessariamente ter arestas associadas a ele, no entanto uma aresta é obrigatoriamente formada por dois vértices, conforme imagem abaixo.

BaseGraph

