

ALGORITMOS E ESTRUTURAS DE DADOS II 2018/2

PROF. FLÁVIO JOSÉ MENDES COELHO

30/09/2018

PROJETO PRÁTICO 1

1 Objetivos

Este **Projeto Prático 1 – PP1**, tem o objetivo de exercitar e avaliar suas habilidades em:

- Codificar as estruturas de dados LISTA, PILHA, FILA, ÁRVORES e GRAFOS (ou outras que você mostre serem eficientes para o problema) na linguagem de programação exigida neste enunciado, e apresentar soluções/algoritmos simples e eficientes para o problema;
- Demonstrar seu domínio sobre o código que você desenvolver neste projeto e mostrar que sabe realizar modificações locais no código do projeto à pedido de um avaliador;
- Apresentar argumentos lógicos, razoáveis, para questões práticas ou teóricas levantadas por um avaliador sobre este projeto.

2 Descrição do problema

A nave estelar U.S.S Enterprise precisa alcançar um posto avançado da Federação Unida dos Planetas. Para isto, o capitão James T. Kirk designou à sua equipe a tarefa de programar a Enterprise para percorrer o **caminho mais curto** entre a posição atual da nave e o posto avançado. Entretanto, a Enterprise deverá explorar cada planeta de cada sistema planetário a ser visitado durante o caminho. A Figura 1 apresenta um possível cenário para o problema e a Figura 2 mostra um grafo representando este cenário. Alguns sistemas planetários são inimigos declarados da Federação, e, portanto, devem ser evitados. Cada sistema planetário é composto por um conjunto de planetas orbitando uma ou mais estrelas; alguns destes planetas são amigos entre si, e por isso, são ligados por rotas de comércio, cooperação tecnológica e intercâmbio cultural (Figura 3). Ao entrar em um sistema, a Enterprise deverá construir uma **árvore de busca em profundidade** iniciando pelo planeta 1 do sistema, escolhendo sempre o planeta amigo de menor numeração. Por exemplo, as árvores de busca em profundidade construídas para os sistemas 9 e 14 da Figura 3, resultam nas listas (1-2, 2-3, 3-4, 4-5, 3-6) e (1-4, 4-2, 2-3, 3-7, 3-8, 8-5, 2-6), respectivamente.

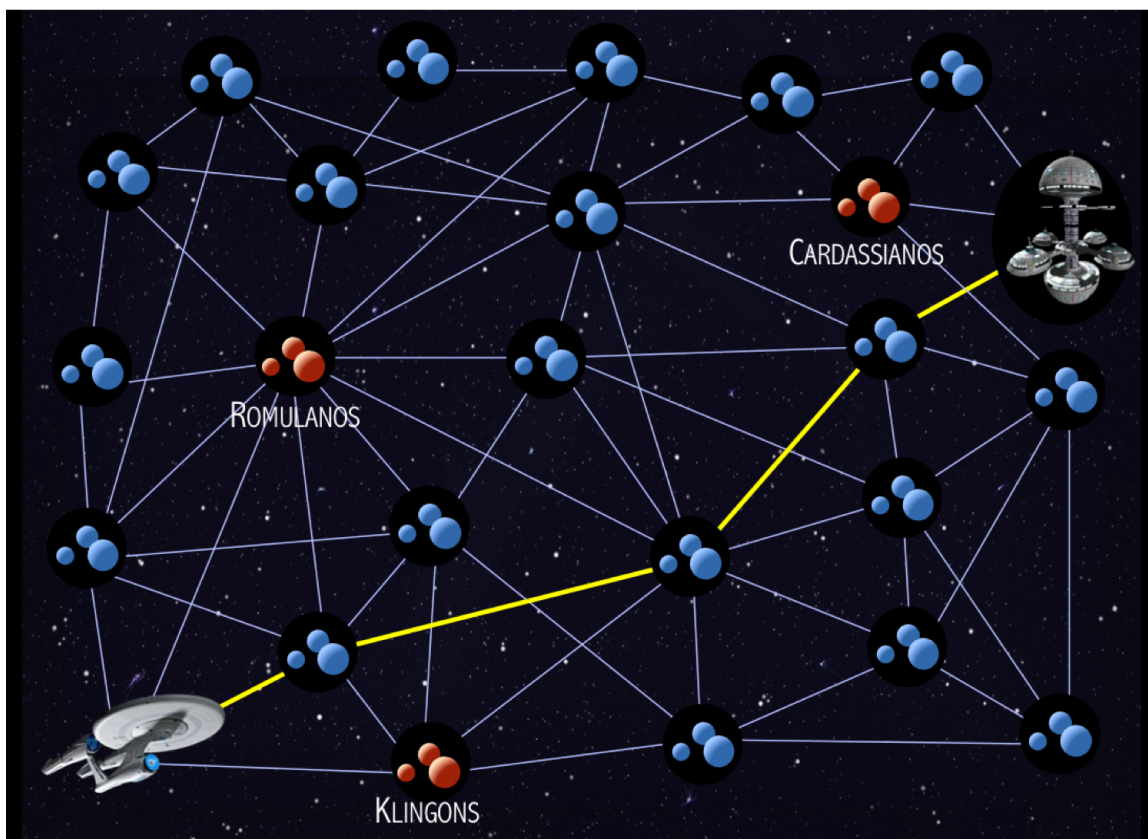


Figura 1: Possível cenário

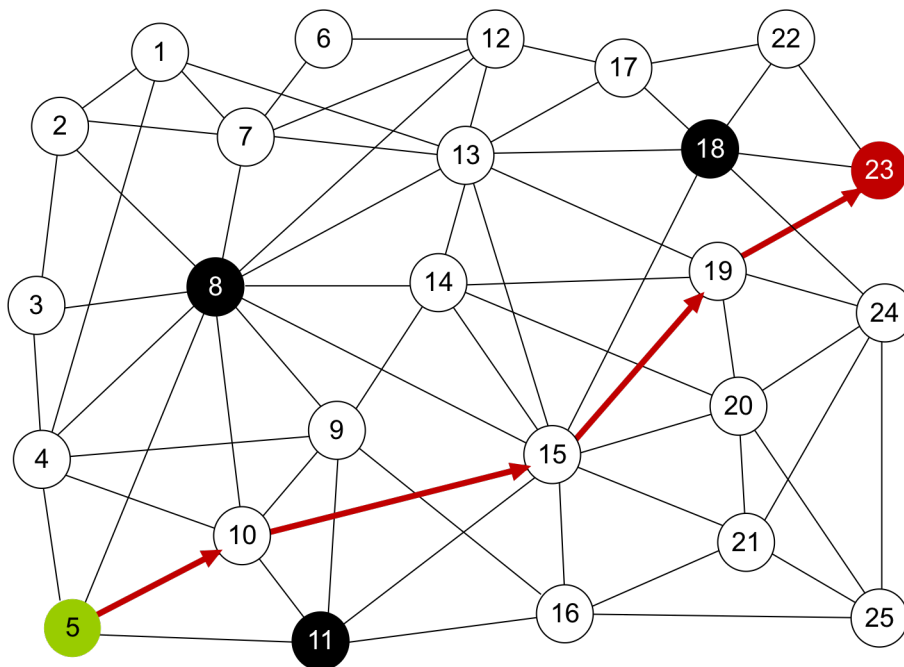


Figura 2: Grafo do cenário da Figura 1

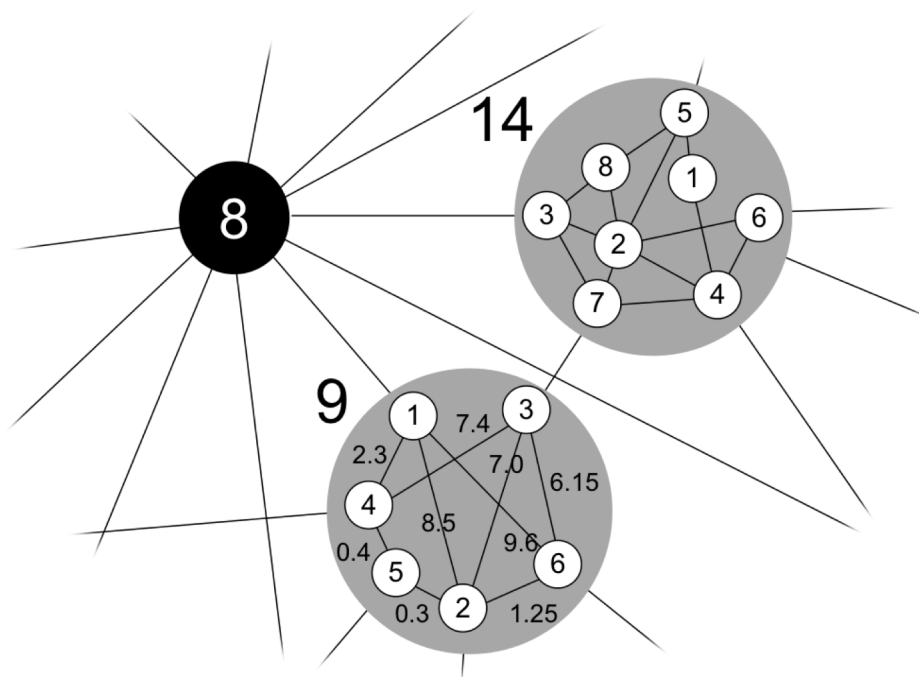


Figura 3: Sistema 9 com suas distâncias entre planetas e sistema 14 do cenário da Figura 1

3 Entradas e saídas do problema

Entradas. As entradas serão lidas a partir de um juiz online, e compreendem as seguintes sequências:

1. O grafo geral da galáxia contendo os sistemas planetários, seguido dos números de sistemas inimigos da Federação, seguido do número do sistema onde a U.S.S Enterprise está inicialmente, seguido do sistema de destino (posto avançado da Federação). Exemplo:

```
25 67 <= ordem e tamanho do grafo geral
1 2   <= 1a. aresta ligando o sistema 1 ao 2
2 8   <= 2a. aresta ligando o sistema 2 ao 8
2 3   <= 3a. aresta ligando o sistema 2 ao 3
3 4   <= 4a. aresta ligando o sistema 3 ao 4
etc...
15 16   <= 67a. aresta ligando o sistema 15 ao 16
3 8 18 11 <= Quantidade de inimigos da Federação seguida dos inimigos
5 23    <= Sistemas inicial e de destino
```

2. Os grafos de cada sistema planetário, na ordem de numeração de cada sistema. Exemplo:

```
6 4   <= ordem e tamanho do grafo do sistema 1
1 3 12.5 <= aresta ligando o planeta 1 ao planeta 3, e sua distância
4 5 9.6  <= aresta ligando o planeta 4 ao planeta 5, e sua distância
5 7 0.3
7 8 1.8
5 6   <= ordem e tamanho do grafo do sistema 2
1 3 1.5 <= aresta ligando o planeta 1 ao planeta 3, e sua distância
1 5 1.9
2 3 7.0
2 4 8.46
2 5 12.66
4 5 2.1
9 8   <= ordem e tamanho do grafo do sistema 3
etc...
6 10 <= ordem e tamanho do grafo do sistema 9
1 4 2.3
1 2 8.5
1 6 9.6
2 5 0.3
2 3 7.0
2 6 1.25
3 4 7.4
3 6 6.15
4 5 0.4
3 3 <= ordem e tamanho do grafo do sistema 10
1 2 5.99
```

1 3 5.4
2 3 5.8
etc...
Segue até o sistema 25

Saída. Deve ser apresentado o total das somas dos custos de cada árvore de busca em profundidade de cada sistema do caminho mínimo percorrido pela U.S.S Enterprise.

4 Requisitos do projeto

1. **Equipes.** Este projeto deve ser desenvolvido por uma equipe de **dois** estudantes. Não serão aceitas equipes com menos de dois ou mais de três participantes. **Importante:** não serão aceitas equipes com número menor ou maior de participantes, sob pena de obter nota 0,0 na parte funcional deste projeto. As exceções devem ser tratadas diretamente com o professor o mais rápido possível, antes da postagem deste enunciado, ou logo após sua postagem.
2. **Ferramentas e técnicas.** O projeto deve ser codificado em C++14 utilizando programação orientada a objetos com encapsulamento, pelo menos nos TADS empregados. Indica-se o uso do compilador *GCC - the GNU Compiler Collection* (<https://gcc.gnu.org>) ou *CLANG* (<https://clang.llvm.org/>), ou compiladores C++ online. Dia: compile seu projeto em vários compiladores online (além do compilador do juiz online) para garantir que o mesmo não apresente problemas sintáticos/semânticos (nem *warnings*). Localmente, aplique as diretivas de compilação mais restritas do seu compilador.
3. **Pontuação.** O **PP1** vale de 0,0 (nota mínima) a 10,0 (nota máxima), e será avaliado em duas fases:

Fase 1 - Avaliação funcional. Pontuação mínima: 0,0; pontuação máxima: 5,0.
O código do projeto será submetido ao juiz online **run.codes**, e obterá a nota máxima desta fase, se passar em todos os casos de teste (cada caso terá uma pontuação correspondente a uma fração da nota máxima desta fase). Os detalhes sobre a submissão serão repassados pelo professor, por e-mail.

Fase 2 - Inspeção de código. Pontuação mínima: 0,0; pontuação máxima: 5,0.
Nesta etapa, o professor escolherá um membro da equipe para defender o projeto (qualquer membro ausente no momento da escolha do membro defensor receberá a nota mínima integral no projeto). Na inspeção de código, serão feitas perguntas sobre detalhes de implementação do projeto de acordo com os seguintes critérios:

- **Legitimidade** (critério eliminatório). A constatação de que o projeto é plágio implica na atribuição automática da nota mínima integral para o projeto incluindo as duas etapas de avaliação. Será considerado legítimo o projeto que tiver similaridade de código entre equipes menor do que 40% (comparador do juiz online do run.codes).
- **Segurança na defesa** ao responder as perguntas do avaliador e explicar as estratégias utilizadas.

- **Cumprimento de requisitos:** o projeto atende ao que foi solicitado neste enunciado.
- **Qualidade de código:** indentação, uso de TADS orientados a objetos (encapsulados), uso do C++14*, programação genérica (quando aplicável), qualidade de código (implementação simples e eficiente das estruturas de dados, boa nomeação de identificadores, escolha das estruturas de dados e algoritmos mais eficientes em desempenho), e criatividade. **Importante:** remova todos os comentários do seu código. **Se o código estiver comentado, a equipe perde o direito à defesa da Fase 2!**
- **Uso de bibliotecas:** Todo o projeto deverá ser submetido ao juiz online em um único arquivo fonte com extensão `cpp`. Sua equipe **não** deve utilizar estrutura de dados `vector`, `list`, etc, ou qualquer outra da *Standard Template Library* - STL, ou de nenhuma outra biblioteca C++ (utilize `iostream`, `cstring`, `cstdlib` e `limits` quando necessário). Portanto, as estruturas de dados (filas, pilhas, grafos, etc) e algoritmos devem ser programadas pela própria equipe. O uso de estruturas de dados prontas de outras bibliotecas C++ ou da STL que não seja as especificadas acima, implicará na atribuição da nota mínima ao projeto.

5 Datas

- Emissão deste enunciado: 30/09/2018 às 17h (hora local).
- Abertura do juiz online: 30/09/2018 - 18:00:00 (hora local).
- Fechamento do juiz online: 14/10/2018 22:59:59 (hora local).
- Inspeção de código: 15/10/2018 de 14h às 20h e 16/10/2018 de 14h às 20h (hora local).

As datas podem ser modificadas para atender demandas da turma ou do professor. Neste caso, as mudanças serão comunicadas ao inscrito no grupo de AED2 por e-mail.

CÓDIGO DE ÉTICA

Este projeto é uma avaliação acadêmica e deve ser concebido, projetado, codificado e testado pela equipe, com base nas referências fornecidas neste enunciado ou nas aulas de Algoritmos e Estruturas de Dados, ou por outras referências indicadas pelo professor, ou com base em orientações do professor para com a equipe, por solicitação desta. Portanto, não copie código pronto da Internet para aplicá-lo diretamente a este projeto, não copie código de outras equipes, não forneça seu código para outras equipes, nem permita que terceiros produzam este projeto em seu lugar. Isto fere o código de ética desta disciplina e implica na atribuição da nota mínima ao trabalho.

Referências

- [1] COELHO, Flávio. Slides das aulas de *Algoritmos e Estruturas de Dados II*. Disponível em <https://est.uea.edu.br/fcoelho>. Universidade do Estado do Amazonas, Escola Superior de Tecnologia, Núcleo de Computação - NUCOMP. Semestre letivo 2018/2.
- [2] C++. In: *WIKIPÉDIA, a enciclopédia livre*. Flórida: Wikimedia Foundation, 2018. Disponível em: <https://pt.wikipedia.org/w/index.php?title=C%2B%2B&oldid=45048480>. Acesso em: 20 jun 2018.
- [3] C++. In: *cppreference.com*, 2018. Disponível em <http://en.cppreference.com/w/>. Acesso em: 20 jun 2018.
- [4] CORMEN, T. H., Leiserson, C. E., Rivest, R. L., Stein C. *Introduction to Algorithms*, 3rd edition, MIT Press, 2010
- [5] KNUTH, Donal E. *Fundamental Algorithms*, 3rd.ed., (vol. 1 de The Art of Computer Programming), Addison-Wesley, 1997.
- [6] KNUTH, Donal E. *Seminumerical Algorithms*, 3rd.ed., (vol. 2 de The Art of Computer Programming), Addison-Wesley, 1997.
- [7] KNUTH, Donal E. *Sorting and Searching*, 2nd.ed., (vol. 3 de The Art of Computer Programming), Addison-Wesley, 1998.
- [8] STROUSTRUP, Bjarne. *The C++ Programming Language*. 4th. Edition, Addison-Wesley, 2013.
- [9] STROUSTRUP, Bjarne. *A Tour of C++*. Addison-Wesley, 2014.
- [10] SZWARCFITER, Jayme Luiz et. alii. *Estruturas de Dados e seus Algoritmos*. Rio de Janeiro. 2a. Ed. LTC, 1994.
- [11] WIRTH, Niklaus. *Algoritmos e Estruturas de Dados*. Rio de Janeiro. 1a. Ed. Prentice - Hall do Brasil Ltda., 1989.
- [12] ZIVIANI, Nívio. *Projeto de Algoritmos com Implementação em Java e C++*. 2a. Edição. Cengage Learning, 2010.
- [13] ZIVIANI, Nívio. *Projeto de Algoritmos com Implementação em Pascal e C*. 3a. Ed. São Paulo: Cengage Learning, 2012.