Universidade Federal do Espírito Santo - Departamento de Informática

Estruturas de Dados I INF09292

1º Trabalho Prático de 2017/01

Prof.ª Patrícia Dockhorn Costa, Email: pdcosta@inf.ufes.br

Data de Entrega: 10/06/2017 - Trabalho em dupla.

Este trabalho tem como objetivo praticar o uso de tipos abstratos de dados e estruturas do tipo Lista.

Regras Importantes

- Não é tolerado plágio. Trabalhos copiados serão penalizados com zero.
- A data de entrega é inadiável. Para cada dia de atraso, é retirado um ponto da nota do trabalho.

Material a entregar

- Impresso: Documentação do trabalho (utilizando as normas ABNT UFES), que deve conter:
    - Capa do Projeto: deve conter os seguintes itens: Título, Autoria e Data.
    - Introdução: descrição do problema a ser resolvido e visão geral sobre o funcionamento do programa (em termos de módulos, arquivos, etc.).
    - Metodologia: descrição da implementação do programa. Devem ser detalhadas as estruturas de dados utilizadas (preferencialmente com **diagramas ilustrativos**), o funcionamento das principais funções utilizadas incluindo pré e pós condições, o formato de entrada e saída de dados, bem como decisões tomadas relativas aos casos e detalhes de especificação que porventura estejam omissos no enunciado. **Modularize o seu programa usando a técnica de tipos abstratos de dados**, como discutido em sala de aula.
    - Conclusão: comentários gerais sobre o trabalho e as principais dificuldades encontradas em sua implementação.
    - Referências bibliográficas: bibliografia utilizada para o desenvolvimento do trabalho, incluindo sites da Internet se for o caso.
- Por email (**estruturasdedadosufes@gmail.com**):
    - O assunto da mensagem deve ser ed20171:trab1:
    - Por exemplo: ed20171:trab1.
    - Documentação do trabalho (em formato PDF).
    - Todos os arquivos .c e .h criados (exigido código muito bem documentado!).
    - O makefile.

Aedes Simulator

O Aedes aegypti é um mosquito vetor de diversas doenças, entre elas Febre Amarela, Zika, Dengue e Chicungunya. É um problema atual no contexto das cidades com clima tropical devido ao alto poder de disseminar doenças e de produzir rapidamente descendentes do mosquito.

A transmissão de doenças é feita através da picada da fêmea do mosquito que possui um determinado vírus. A pessoa que é picada contrai a doença e, caso outro Aedes aegypti pique a mesma pessoa, o mosquito vira um novo vetor com o vírus da doença. Além disso, o mosquito passa o vírus entre as gerações, ou seja, cada descendente é um vetor com o vírus desde o nascimento. A fêmea do Aedes aegypti necessita do sangue humano para fazer a oviposição em água. Os ovos, então, eclodem e viram larvas que se alimentam da matéria orgânica presente na água. Em seguida, as larvas viram pupas, que por sua vez dão origem a novos descendentes.

O principal modo para o combate das doenças transmissíveis pelo Aedes aegypti é a prevenção do nascimento de mais mosquitos: não deixar água parada em utensílios e evitar o acúmulo de água da chuva. A prevenção é aliada aos agentes de saúde municipais, responsáveis por educar a população através de visitas domiciliares preventivas. Também, as prefeituras de diversas cidades brasileiras costumam utilizar o fumacê, para diminuir uma população local de mosquitos. As armadilhas para o Aedes aegypti são importantes indicadores da população de mosquitos em uma determinada região e uma outra importante forma de combate.

O Aedes Simulator, proposto por este trabalho, é um simulador de comportamento do Aedes aegypti no ambiente urbano. O objetivo do simulador é prever um possível cenário de epidemia em uma região, melhorando o entendimento do hábito de vida do mosquito e permitindo o planejamento de ações mais efetivas.

O simulador deste trabalho representa o ambiente de uma região. As casas são ligadas de forma a representar a possibilidade de vôo do mosquito de uma casa à outra. Os mosquitos inseridos podem mudar sua posição ao longo do tempo. Os agentes atuam de forma a escolher uma casa e acabar com os mosquitos. Por questões didáticas, a simulação foi simplificada. Em uma modelagem mais realista, outros elementos, tais como seres humanos, comportamentos reais do mosquito, vírus e clima, poderiam ser incluídos para torná-la mais acurada.

1ª Parte: Implementação da Estrutura de Dados

Para a simulação do cenário, é necessária a implementação da estrutura de dados conforme a "entrada", indicando as entidades que farão parte do processo. Faça a leitura do arquivo "entrada.txt" que contém os comandos e transforme os elementos do cenário em uma lista de lista encadeada.

Exemplo de arquivo de entrada:

AGENTE_ATUA 2

MOSQUITO_BOTA 5

inserecasa C1

inserecasa C2

inserecasa C3

ligacasas C1 C2

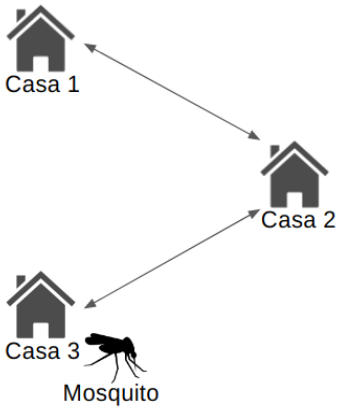
ligacasas C2 C3

inseremosquito C3

iniciasimulacao 50

FIM

Os comandos apresentados equivalem ao cenário demonstrado abaixo:



Especificação dos Comandos:

AGENTE_ATUA <movimentos> : Especifica o número de movimentações dos mosquitos (ao todo) antes de cada chamada da função agente_atua().

MOSQUITO_BOTA <movimento> : Especifica o número de movimentação do mosquito (cada mosquito) antes de cada chamada da função mosquito_bota().

inserecasa <nome_casa> : Insere uma casa de um determinado nome na simulação.

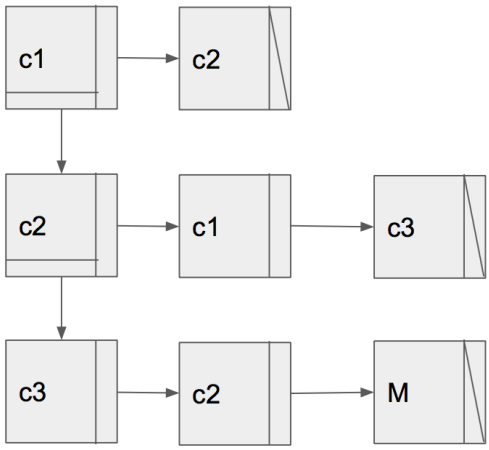
ligacasas <casa_1><casa_2> : Insere uma ligação entre uma casa e outra que permite o vôo do mosquito. A ligação é sempre bidirecional.

inseremosquito <nome_casa> : insere o mosquito em uma determinada casa.

iniciasimulacao <movimentos> : inicia a simulação limitando os movimentos dos mosquitos (ao todo) a um determinado número.

FIM : finaliza a simulação, gerando o arquivo de log e limpando a memória.

O exemplo anterior deve gerar uma lista de listas, como a da figura a seguir:



2ª Parte: Implementação da Simulação

Na 1ª parte, o cenário da simulação foi implementado. Nesta nova parte, o cenário deve ser implementado de forma a ganhar vida. Para tanto, três funções devem ser implementadas: **mosquito_move()**, **agente_atua()** e **mosquito_bota()**.

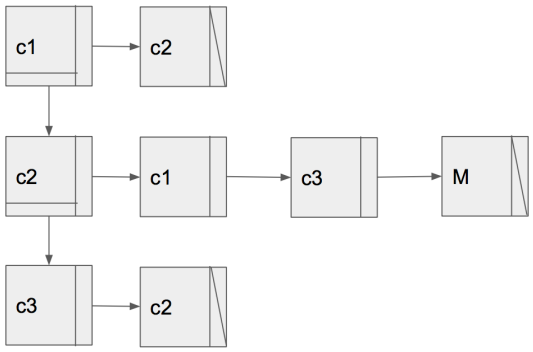
Estas funções são chamadas durante o comando **iniciasimulacao <movimentos>**. O número de movimentos limita a movimentação ao todo dos mosquitos. Observe que é a soma da movimentação de todos os mosquitos.

As funções são explicadas a seguir:

mosquito_move()

Quando chamada, cada mosquito do cenário deve mudar a sua localização para uma casa que possui ligação com sua atual casa. A escolha da nova casa é aleatória (desde que exista ligação direta).

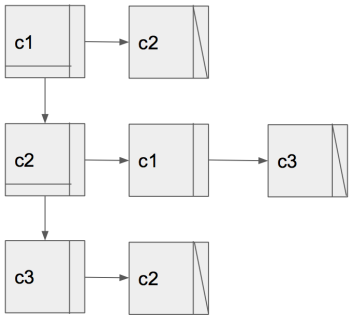
Exemplo: Mosquito escolheu nova casa. A figura seguinte representa este comportamento em comparação ao cenário inicial. Na figura em questão, o mosquito está na casa 2.



agente_atua()

Quando chamada, uma casa é sorteada e há o combate ao mosquito. Isto é, se houver mosquitos na casa sorteada, eles devem sair da lista. A função agente_atua() deve ser chamada após as movimentações ao todo dos mosquitos, ou seja, no exemplo da entrada acima (AGENTE_ATUA 2), os mosquitos serão combatidos após 2 chamadas da função mosquito_move() para cada mosquito.

Exemplo: No caso da entrada acima, que especifica 2 chamadas, se houver 1 mosquito, 2 movimentações devem ser executadas antes da chamada. Se houver 2 mosquitos, 4 movimentações devem ser executadas (2 para cada) antes do agente_atua(). Se houver 3 mosquitos, 6 movimentações devem ser executadas (2 para cada) antes do agente_atua() e assim por diante. Se em uma determinada movimentação o agente "acertar" o local do mosquito, o nó correspondente ao mosquito deve ser retirado da estrutura de dados. Caso "erre" o local, o mosquito continuará da mesma forma. A figura a seguir representa uma correta atuação do agente na casa 3, a partir do cenário inicial apresentado no começo do trabalho.

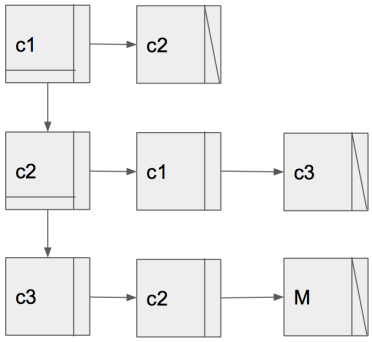


mosquito_bota()

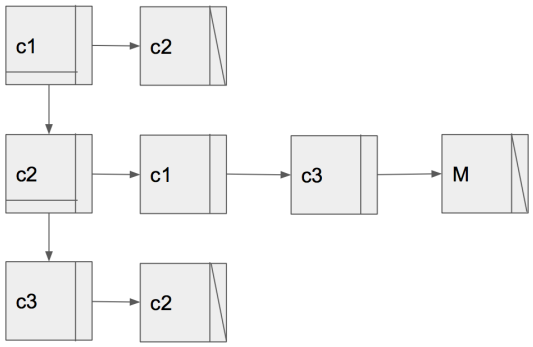
Quando chamada, o mosquito cria mais dois descendentes diretos. A função deve ser chamada após os movimentos de cada mosquito. No caso da entrada indicada, cada mosquito terá descendentes após 5 movimentações (MOSQUITO_BOTA 5).

Exemplo: Utilizando a entrada que especifica 5 movimentações, caso um único mosquito possa se movimentar 5 vezes na simulação, dois outros mosquitos são criados após a estrutura do mosquito em questão. A figura a seguir apresenta o comportamento de 5 movimentações e dois novos mosquitos.

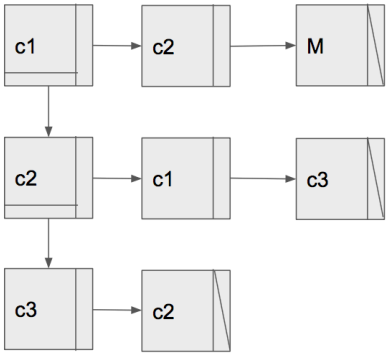
Cenário Inicial:



Primeira movimentação (mosquito_move() para a C2):

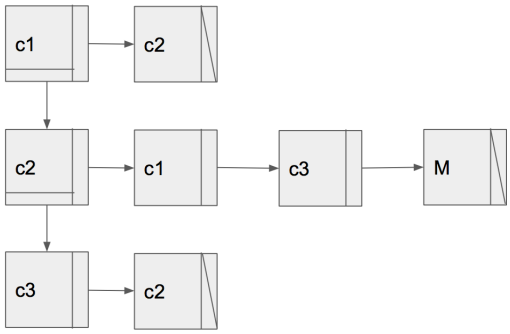


Segunda movimentação (mosquito_move() para a c1):

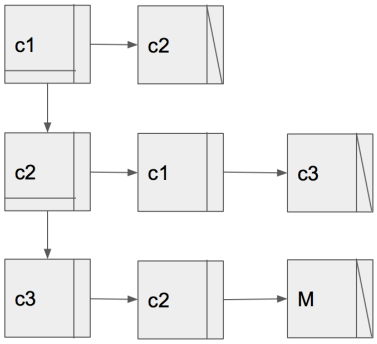


(observe que após a segunda movimentação há uma chamada para agente_atua(). Isso será desconsiderado nesse exemplo)

Terceira movimentação (mosquito_move() para a c2):

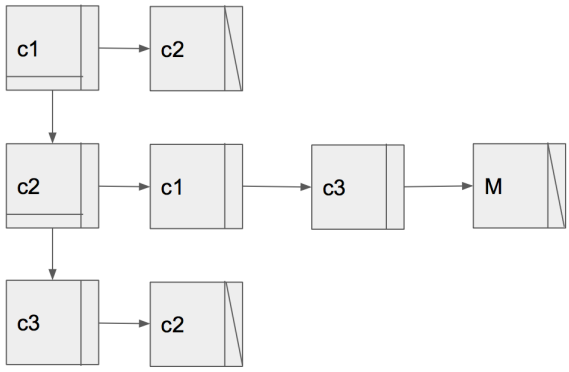


Quarta movimentação (mosquito_move() para a c3):

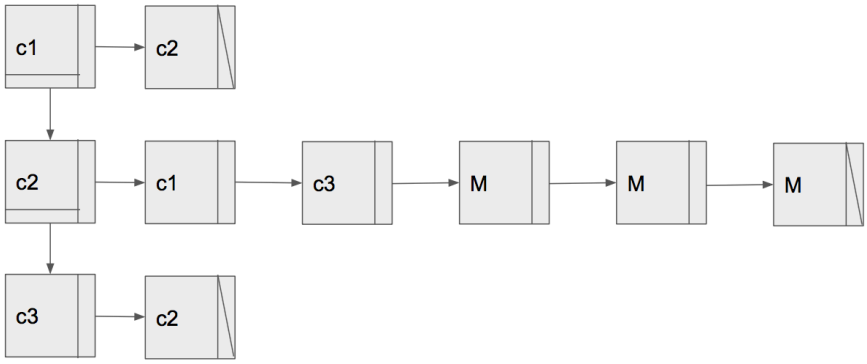


(observe que após a quarta movimentação novamente há uma chamada para agente_atua() que também será desconsiderada)

Quinta movimentação (mosquito_move() para a c2):



Após a quinta movimentação, há a chamada para mosquito_bota():



3ª Parte: Implementação do Log

Ao final da execução da simulação, o programa deve ter gravado informações à respeito da simulação em um arquivo log.txt:

Número de mosquitos iniciais:

Número de mosquitos finais:

Número de movimentos totais dos mosquitos:

Número de erros dos agentes:

Número de acertos dos agentes:

Além do log.txt, o programa deve gravar a lista inicial e final no lista.txt, como no exemplo das figuras anteriores (figura de cenário inicial e figura mosquito_bota):

Inicial:

c1 -> c2

c2 -> c1 -> c3

c3 -> c2 -> M

Final:

c1 -> c2

c2 -> c1 -> c3

c3 -> c2 -> M -> M -> M

Atenção: A saída do programa é variada pois o programa possui uma execução aleatória. Portanto, dada uma determinada entrada, a saída poderá se diferenciar.

Observações importantes:

- Caso o mosquito seja combatido antes do número de movimentações, isso é, se não houver mais nenhum mosquito na memória, a execução deve ser terminada e deve haver a escrita do log.txt .
- Observe o número de movimentações: o número de movimentações necessárias do mosquito para a execução do agente_atua e mosquito_bota é contabilizado de forma diferente: enquanto agente_atua especifica as movimentações gerais (movimentação de todos os mosquitos), mosquito_bota especifica movimentações de cada mosquito (a regra é contabilizada por mosquito).
- Podem existir vários mosquitos, tome cuidado com esse caso.
- Sempre haverá uma ligação de uma casa a outra. A ligação é de via dupla, isso é, se C1 vai para C2, C2 vai para C1.
- Cada casa contém um nome diferente, que permite identificá-la.
- Os mosquitos sempre ficam ao final da lista encadeada.
- Os mosquitos contêm um inteiro para o controle da movimentação. **Evite ao máximo usar variáveis globais**!
- As funções propostas são as principais funções, não são as únicas a serem implementadas!

Bom trabalho!!!