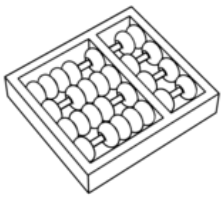

Instituto de Computação
Universidade Estadual de campinas

MC202 - Estruturas de Dados

Laboratório 2

Listas ligadas

Data de publicação: Sexta feira, 18 de março de 2016
Prazo máximo de submissão: Sexta feira, 8 de abril de 2016 às 23h59m

Professor: Neucimar J. Leite <neucimar@ic.unciamp.br>

Monitores:
- Juan Hernández (PED) <juan.albarracin@students.ic.unicamp.br>
- Leonardo Yvens (PAD) <leoyvens@gmail.com>

Grupo do curso: https://groups.google.com/d/forum/mc202bc_2016s1
Sítio eletrônico da submissão do código: https://susy.ic.unicamp.br:9999/mc202bc

Enunciado

Simule com listas ligadas em C o corte de moléculas circulares de DNA por enzimas de restrição e implemente a busca de sequências específicas após o corte.

Descrição do problema

Um dos problemas mais populares em biologia molecular computacional é o casamento de subsequências da molécula de DNA com um ou vários padrões específicos, para determinar se eles estão contidos na molécula.
Em ocasiões, o análise de genomas requer que as moléculas de DNA sejam divididas longitudinalmente. Isto tem inúmeras aplicações. As enzimas de restrição são utilizadas para cortar as moléculas em regiões específicas, chamadas de **regiões de restrição**. Elas reconhecem uma sequência de nucleotídeos e executam o corte segundo um padrão determinado. Segue um exemplo:

Seja a molécula de DNA

```
a c t a a g t c c
| | | | | | | | |
t g a t t c a g g
```

E a enzima de restrição que reconhece a sequência **a c t** e tem o padrão de corte **a c | t**, a molécula seria cortada em dois fragmentos, como mostrado abaixo:

```
a c             t a a g t c c
| |                   | | | |
t g a t t           c a g g
```

Neste ponto, é preciso relembrar alguns conceitos básicos sobre a molécula de DNA:

- Ela está formada por duas cadeias entrelaçadas que são **antiparalelas**, isto é, cada cadeia tem um sentido específico e o sentido de uma sempre é o contrário da outra. Por exemplo, na molécula acima, a cadeia superior se lê `actaagtcc`, já a cadeia inferior se lê `ggacttagt`. É por isto que a enzima de restrição fez o corte naquela região da cadeia inferior.
- As letras na molécula se chamam de **nucleotídeos**. Neste laboratório vamos trabalhar com os nucleotídeos mais comuns, que são **a**, **c**, **g** e **t**. O casamento dos nucleotídeos de uma cadeia com os nucleotídeos da outra não é arbitrário: o **a** só poder ser casado com **t**, e o **c** com **g**. O casamento de dois nucleotídeos se chama de **base nitrogenada**.

Após o corte, torna-se necessário localizar alguma subsequência e determinar em quais fragmentos mantem-se uma base nitrogenada com ela. No exemplo acima, a sequência **ac** (ou **tg**, note) encontra-se em ambos os fragmentos, mas a sequência **ta**, embora possa ser lida em ambas cadeias, não forma parte de uma base nitrogenada, sendo que a enzima de restrição destruiu os enlaces, e portanto não foi mantida.

Atividade a ser feita

Neste laboratório, visando a pôr em prática a maior quantidade de conceitos referentes a listas ligadas, e para simplificar o problema, vamos ter três supostos:

1. As moléculas de DNA são circulares. Portanto, a cadeia superior da molécula acima contem a sequência **cca**, e após um corte, ela ainda tem um único fragmento que agora é linear.

2. O padrão de corte duma enzima de restrição pode apagar os dois nucleotídeos adjacentes ao ponto de corte. Portanto, se esse for o caso no exemplo acima, os fragmentos resultantes seriam:

```
a           a a g t c c
|                 | | |
t g a t           a g g
```

3. Uma enzima lê cada cadeia em ordem e só pode fazer um corte por cadeia. Portanto, se houver mais de uma região de restrição na mesma cadeia, o corte será feito unicamente na primeira região de restrição encontrada de cada cadeia.

4. As bases nitrogenadas que são destruídas, são aquelas que pertencem à sequencia que começa no ponto de corte da primeira cadeia e termina no ponto de corte na segunda. A lógica consiste em seguir a ordem de leitura das cadeias.

O seu programa deve receber a cadeia superior da molécula de DNA, a região de restrição com seu respetivo padrão e a subsequência a ser buscada. Dado que o corte fará com que a molécula fique linear, o programa deve reorganizar a molécula para poder associar índices ao nucleotídeos. O critério para isto será fazer que os extremos da molécula fiquem no principio ou final da representação e que a ordem de leitura da cadeia superior seja mantida de esquerda a direita. O programa deve indicar se a subsequência está contida na molécula (seja a cadeia superior, seja a cadeia inferior) e em quais posições, lendo a molécula de esquerda a direita.

Será fornecido o arquivo **list.h** com a declaração de seis TADs:

- typedef struct SinglyLinkedListNode SinglyLinkedListNode;
- typedef struct DoublyLinkedListNode DoublyLinkedListNode;
- typedef struct SinglyLinkedList SinglyLinkedList;
- typedef struct CircularSinglyLinkedList CircularSinglyLinkedList;
- typedef struct DoublyLinkedList DoublyLinkedList;
- typedef struct CircularDoublyLinkedList CircularDoublyLinkedList;

Que correspondem a quatro tipos de listas (simplesmente ligada, duplamente ligada, simplesmente ligada circular, duplamente ligada circular) e dois tipos de nós (com um e com dois ponteiros). Adicionalmente, para cada tipo de lista, serão fornecidos 5 cabeçalhos de funções onde devem ser implementadas as seguintes operações:

- Criar lista
- Destruir lista
- Inserir elemento
- Apagar elemento
- Atualizar elemento

Certamente, você pode concluir que nem todos os tipos de listas e operações serão necessários para resolver o problema deste laboratório, logo tem a liberdade de implementar unicamente os TADs das listas que vai usar, sempre que **justifique brevemente em comentários dentro do código** por que não usará as demais listas e operações. Não é permitido o uso de outras estruturas. Caso deseje implementar

operadores adicionais sobre as estruturas, pode declarar, porém deve justificar a utilidade do operador em comentários.

Exemplo detalhado

Molécula	Região de restrição e padrão de corte	Subsequência a ser buscada
t t c c a g a a t g t a | | | | | | | | | | | | a a g g t c t t a c a t	a t t | c	a a t

Após o corte, a molécula fica:

```
t t     c c a g a a t g t a
| |             | | | | | |
a a g g t c     t t a c a t
```

Vamos supor que na entrada foi especificado que os nucleotídeos adjacentes ao ponto de corte devem ser apagados.

```
t         c a g a a t g t a
|                 | | | | |
a a g g t         t a c a t
```

Sendo que precisamos atribuir posições à molécula para saber onde pode se encontrar a subsequência, vamos reorganizá-la para os extremos ficarem no princípio ou final da representação. Isto é:

```
0  1  2  3  4  5  6  7  8  9  10 11 12 13
c  a  g  a  a  t  g  t  a  t
          |  |  |  |  |  |
          t  a  c  a  t  a  a  g  g  t
```

Os números em azul são os índices da nova representação. Note que agora os extremos da molécula conferem com os extremos da representação, e que a ordem de leitura da cadeia superior é ainda de esquerda a direita. Isto faz com que as bases nitrogenadas (neste caso, a região entre 4 a 9) fiquem sempre no centro da representação.

Agora, buscando a subsequência **`a a t`**, podemos ver que ela está na região entre 3 e 5 (lendo a cadeia superior) e na região entre 8 e 10 (lendo a cadeia inferior). Porém, o primeiro nucleotídeo nas duas regiões não pertence a uma base nitrogenada (ao não estar

casado com um nucleotídeo da outra cadeia), portanto não podemos dizer que a subsequência existe.

Se você compreendeu corretamente como funciona o corte, concordará em que, se os nucleotídeos adjacentes ao ponto de corte não tivessem sido apagados, a subsequência seria mantida na região entre 3 e 5 e na região entre 8 e 10.

Especificação de entrada e saída

Para ser avaliado, o programa desenvolvido deve fazer leitura dos dados de entrada e a posterior escrita dos resultados. Segue a estrutura da entrada ao programa:

- A primeira linha contém a cadeia superior da molécula, sem espaços.

- A segunda linha contém três informações separadas por espaço. A primeira, é a região de restrição que deve ser reconhecida pela enzima, sem espaços.

  Após o primeiro espaço está o índice de corte, isto é, o ponto na região de restrição onde o corte será feito.

  Após o segundo espaço está um valor que indica se haverá ou não apagamento dos nucleotídeos adjacentes ao ponto de corte: 0 indica que não haverá, 1 indica que haverá.

  Quanto o índice de corte, segue um exemplo com uma região de comprimento 6:

  $$_0a_1c_2g_3t_4a_5g_6$$

  O valor do índice vai desde 0 até *l*, onde *l* é o comprimento da região. No exemplo acima, um índice de corte 3 significa o padrão **`acg|tag`**.

- A terceira linha contém a subsequência a ser buscada na molécula após o corte.

A saída deve ser uma lista de números inteiros indicando a posições na molécula linear onde estão as bases nitrogenadas de menor índice de cada subsequência encontrada, ou as letras **NE** para indicar que a subsequência não está na molécula. Caso estejam sobrepostas, e.g. buscar **`gtgt`** e a molécula contém **`gtgtgt`**, deve se indicar os índices de todas as sequências encontradas, neste caso, 0 e 2.

Seguem alguns exemplos de entradas com suas respectiva saídas.

Entrada	Saída	Comentários

`acgggttatt`
`ta 1 1`
`cgg`
	`3`	O corte na primeira cadeia é feito à direita da posição 7 (contando desde 0); na segunda cadeia, o corte é feito também à direita de 7. Lembre-se que a entrada pede que os

		nucleotídeos adjacentes ao ponto de corte sejam apagados, portanto as posições 7 e 8 são descartadas em ambas as cadeias. Temos que recalcular os índices para deixar o extremo esquerdo da molécula na posição 0. Então, a nova representação começa com o t da posição 9 da molécula original.

`acgggttatt`
`ta 1 0`
`cgg`
	`4`	Neste caso, os nucleotídeos ficam na molécula, portanto haverá mais um no princípio da molécula.

`ttccagaatgta`
`attc 3 0`
`aat`
	`4 9`	Entrada correspondente ao exemplo detalhado acima. As duas regiões que contêm a subsequência vão de 4 a 6 e de 9 a 11. Os menores índices de cada uma são, respectivamente 4 e 9.

`ttccagaatgta`
`attc 3 1`
`aat`
	`NE`	

`ccaagg`
`gg 1 0`
`ca`
	`2`	O corte na primeira cadeia fica após 4, e o corte na segunda cadeia fica após 0. As bases nitrogenadas que são separadas, estão em 5 e 0. Após reorganizar a molécula, a cadeia superior fica gccaag. ca começa em 2 e formam bases nitrogenadas.

Estrutura da submissão

O código fonte deve ser submetido no sistema SuSy para ser executado e testado. O sistema receberá três arquivos:

Nome	**Função**
main.c	Programa principal que lê os dados de entrada, faz o chamados às funções e escreve a saída.
list.h	Arquivo com os cabeçalhos e declarações descritos acima. Esse arquivo será fornecido inicialmente e, embora não seja permitido modificar a estrutura dos cabeçalhos presentes, permite-se acrescentar cabeçalhos de outras funções, **sempre que sejam operações referentes a listas**. Funções que não tenham nada a ver com listas, podem ser implementadas no arquivo **main.c**.
list.c	Código fonte das funções especificadas em **oper.h**.

Observações

- É mandatório liberar memória dinámicamente alocada.

- Arquivos de teste serão fornecidos para o estudante validar seu programa. A avaliação no sistema será feita com esses e outros testes privados.

- O limite de submissões no SuSy é 15. Recomenda-se executar de forma local seu programa, usando tanto os testes abertos, quanto os exemplos neste documento para avaliar a o desempenho do mesmo. Unicamente assim que o programa tiver resolto corretamente todos os casos, é hora de submeter.

- A partir deste laboratório, a claridade do código e a sua documentação (por meio de comentários) serão avaliadas. O esforço dos monitores para entender o seu código será mínimo. Com a documentação, você demonstra que compreendeu de maneira efetiva os conteúdos do curso referente ao tema de listas ligadas.

- Dúvidas podem ser esclarecidas nas aulas de laboratório ou no grupo do curso indicado no cabeçalho deste documento.

Critério de avaliação

$$nota = 5\frac{n_c}{n} + QD + AC$$

$$0 \leq QD \leq 2$$
$$0 \leq AC \leq 3$$

onde,

n	:	Número total de testes no SuSy
n_c	:	Número de testes corretos
QD	:	Qualidade do código
AC	:	Indicador de quanto o estudante tem demostrado dominar o tema de listas ligadas

A qualidade do código (*QD*) dependerá tanto da legibilidade quanto da documentação. Deve ficar fácil para os monitores entender o que foi feito. Já o *AC* é uma nota que depende de quanto o estudante conseguiu convencer ao revisor de que sabe implementar corretamente os TADs, identificar quais estruturas de dados são as mais apropriadas para resolver o problema e operar com essas estruturas.

Considerações finais

- Embora existam várias maneiras de resolver os problemas indicados nos laboratórios, o estudante deve optar pela maneira que melhor aplique os conceitos trabalhados em sala de aula.
- Casos de plágio acarretam **média final zero** para todos os envolvidos, sem exceção. O SuSy pode detectar casos de plágio no código, portanto evite compartilhar seu código com outros estudantes, mesmo que seja apenas um trecho.