SME0230 - Introdução à Programação de Computadores

PRIMEIRO SEMESTRE DE 2016

Professora: Marina Andretta (andretta@icmc.usp.br)

Monitores: Amanda Carrijo Viana Figur (amanda.figur@usp.br)
Kleber Roberto Stamboni (kleber.stamboni@usp.br)
Vinicius Volponi Ferreira (vinicius.volponi.ferreira@usp.br)

Exercício de Laboratório 9 - Bônus

13/05/2016

Data máxima de entrega: **10/06/2016 até às 23h59**. Este exercício será um bônus. Na disciplina haverá outros 10 conjuntos de exercícios a serem entregues. A média será calculada somando-se a nota dos **11** conjuntos de exercícios e dividindo-a por **10**.

Forma de entrega: Os exercícios deverão ser enviados por e-mail para

`exercicios.sme0230.2016@gmail.com`

O assunto do e-mail deverá ser `IPC_Lab9`. Todos os exercícios devem estar em um único arquivo `zip` com o seguinte nome `IPC_Lab9_`<*número usp*>.

Formato dos arquivos: No início de cada arquivo deve haver um comentário com o nome e o número USP do aluno.

Para cada algoritmo, o nome do arquivo deverá ser

`Ex`<*i*>`_`<*número usp*>`.c`, em que <*i*> representa o número do exercício correspondente.

Exemplo
`Ex1_123456.c`

Observações importantes:

- Trabalhos entregues fora do prazo não serão aceitos.
- É muito importante que seu programa tenha comentários e esteja bem indentado, ou seja, digitado de maneira a ressaltar a estrutura de subordinação dos comandos do programa. A avaliação dos exercícios levará isto em conta.
- Cada programa deve ser executado tantas vezes quantas forem necessárias para testar todos os casos possíveis para as entradas.

Dica: Para criar um arquivo `zip` no Linux, basta digitar no terminal

`zip` <*arquivo de saída*>`.zip` <*arquivos de entrada*>

Exemplo
`zip IPC_Lab9_123456.zip Ex1_123456.txt Ex2_123456.c`

Exercício 1

(Baseado em `https://projecteuler.net/problem=400`.) Uma **Árvore de Fibonacci** é uma árvore binária recursivamente definida como:

- T_0 é a árvore vazia.
- T_1 é a árvore unitária (um único nó)
- T_k é o nó raiz que tem os nós T_{k-1} e T_{k-2} como filhos

Numa árvore dessas, dois jogadores, P_1 e P_2, jogam um jogo de retirar nós. A cada turno um jogador seleciona um nó e o retira junto com a subárvore enraizada no nó escolhido. O jogador que retirar o nó raiz, isto é, o primeiro nó da árvore, **perde** o jogo.

Definimos $f(k)$ como o número de possíveis jogadas iniciais que o jogador P_1 pode fazer de forma que ele garanta sua vitória (ou seja, com essas jogadas, não há modo de o jogador P_2 vencer) ao jogarem na árvore T_k.

Faça um programa, em linguagem C, que leia um inteiro $k > 0$ e calcule $f(k)$.

Observações:

- Use alocação dinâmica (não se esqueça do `free`!!).
- Modularize seu código.

Dica 1. *Se não entender as dicas abaixo, ignore-as. Saiba também que elas contém spoilers para resolver o exercício.*

Dica 2. *Uma árvore T_k sempre tem k níveis (uma fileira horizontal de nós).*

Dica 3. *Preste atenção na paridade da quantidade de nós presente em cada nível da árvore.*

Dica 4. *Pode ser útil saber a paridade de um conjunto específico de níveis da árvore.*

Dica 5. *Seria interessante definir uma struct para representar um nó e seus (até dois) filhos.*

A próxima página contém uma breve ilustração da estratégia vencedora de P_1.

Abaixo (em vermelho) estão destacadas as escolhas vitoriosas de P_1 para as árvores de T_1 a T_6. Note que $f(5) = 1$ e $f(6) = 3$, por exemplo. Perceba que no caso T_1, P_1 sempre perde.