

UNIVERSIDADE FEDERAL DE MINAS GERAIS
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

DCC004 – Algoritmos e Estruturas de Dados II

Prof. Cristiano Arbex Valle e Gisele L. Pappa Monitor: Fernando Akio

TRABALHO PRÁTICO 2
PREPARAÇÃO PARA A COPA DO MUNDO

Valor: 25 pontos
Entrega: 01/07/2018

1 Descrição

Neste trabalho, vocês irão receber um arquivo contendo o histórico completo de jogos internacionais entre seleções, desde 1872 até 2018. Abaixo, um exemplo do formato do arquivo:

```
1958-06-24,Brazil,France,5,2,FIFA World Cup,Solna,Sweden,TRUE
1958-06-24,Sweden,Germany,3,1,FIFA World Cup,Göteborg,Sweden,FALSE
1958-06-28,France,Germany,6,3,FIFA World Cup,Göteborg,Sweden,TRUE
1958-06-28,German DR,Poland,1,1,Friendly,Rostock,German DR,FALSE
1958-06-29,Denmark,Norway,1,2,Nordic Championship,Copenhagen,Denmark,FALSE
1958-06-29,Sweden,Brazil,2,5,FIFA World Cup,Solna,Sweden,FALSE
1958-07-02,Singapore,Indonesia,0,5,Friendly,Kallang,Singapore,FALSE
1958-07-03,Malaysia,Indonesia,2,4,Friendly,Kallang,Singapore,TRUE
1958-08-16,New Zealand,Australia,2,3,Friendly,Wellington,New Zealand,FALSE
1965-06-30,Sweden,Brazil,1,2,Friendly,Solna,Sweden,FALSE
1978-06-03,Brazil,Sweden,1,1,FIFA World Cup,Mar del Plata,Argentina,TRUE
1982-06-15,Norway,Denmark,2,1,Nordic Championship,Oslo,Norway,FALSE
```

Cada linha é separada por vírgula e contém: data, equipes participantes, placar, razão do jogo e local (cidade, país) onde ocorreu. A última coluna é TRUE caso o jogo não tenha sido em campo neutro. Porém, não utilizaremos todos estes campos.

Neste trabalho, processaremos este arquivo para obter as seguintes informações:

- um ranking que compara a performance histórica dos países em Copas do Mundo,
- uma lista ordenada por frequência dos confrontos. Isto é, vamos descobrir quais são historicamente os jogos mais comuns entre países.

A montagem do ranking deve levar em consideração os critérios abaixo. Ordenamos pelo primeiro critério. Em caso de valores iguais, desempatamos pelo segundo critério, continuando desta forma até o último critério:

1. Número de pontos. Vitórias, empates e derrotas valem 3, 1 e 0 pontos respectivamente. Em caso de empate,
2. Aproveitamento de pontos, definido como $100 \times \text{Pontos} / (3 \times \text{Jogos})$.
3. Saldo de gols (gols marcados menos gols sofridos),
4. Número de gols marcados,

5. Ordem alfabética.

Cada ranking deve incluir pontos, número de jogos, vitórias, empates, derrotas, gols marcados, gols sofridos, saldo de gols e aproveitamento (este último com duas casas decimais).

Já a lista de confrontos deve simplesmente mostrar os dois países e o número de confrontos entre eles, e deve ser ordenada do mais frequente para o menos frequente. Em caso de empate, a lista deve ser ordenada pela ordem alfabética dos países.

No exemplo acima, há 5 jogos de Copa do Mundo e 12 jogos no total. Neste caso, a saída seria:

```
Ranking,P,J,V,E,D,GM,GS,SG,AP
Brazil,7,3,2,1,0,11,5,6,77.78
Sweden,4,3,1,1,1,6,7,-1,44.44
France,3,2,1,0,1,8,8,0,50.00
Germany,0,2,0,0,2,4,9,-5,0.00
```

Confrontos

```
Brazil,Sweden,3
Denmark,Norway,2
Australia,New Zealand,1
Brazil,France,1
France,Germany,1
German DR,Poland,1
Germany,Sweden,1
Indonesia,Malaysia,1
Indonesia,Singapore,1
```

Observe que nos confrontos o primeiro país é o que vem antes na ordem alfabética.

Ao ler o arquivo, você deverá guardar os países e os confrontos em alguma(s) estrutura(s) de dados. Deve-se buscá-los na estrutura para atualizar seus dados, caso já tenham aparecido. Devem ser implementados três métodos de busca com três complexidades de caso médio diferentes:

1. Inserção $O(1)$ e pesquisa $O(n)$,
2. Inserção em $O(\log n)$ e pesquisa $O(\log n)$.
3. Inserção $O(1)$ e pesquisa $O(1)$.

A ordenação dos rankings deve ser feita com métodos cuja complexidade de caso médio é no máximo $O(n \log n)$.

O programa deve receber dois argumentos de linha de comando obrigatórios e dois opcionais: o arquivo de entrada, o método de busca a ser utilizado (1, 2 ou 3 de acordo com os casos acima), o ano inicial e o ano final. Por exemplo:

```
./tp2 results.csv 2 1950 1960
./tp2 results.csv 1 1950
./tp2 results.csv 3
```

No primeiro caso, utilizaremos busca em $O(\log n)$ e incluiremos apenas jogos que ocorreram entre 1950 e 1960 (inclusive). No segundo caso, apenas jogos que ocorreram a partir de 1950 e busca $O(n)$. No terceiro caso, todos os jogos devem ser incluídos e a busca deve ser $O(1)$.

Na documentação, deve-se incluir uma tabela comparando o tempo de execução do programa com cada método de busca diferente. Esta comparação deve ser para o caso onde todos os jogos são incluídos.

Pontos extras serão atribuídos se você montar também o Ranking FIFA de seleções. O ranking é feito uma vez por mês e leva em consideração os jogos dos 4 anos anteriores, dando pesos diferentes para competições diferentes e para amistosos. A pontuação de um jogo depende também da posição do oponente no ranking imediatamente anterior. Para mais detalhes sobre o cálculo, acesse:

<http://www.fifa.com/fifa-world-ranking/procedure/men.html>

Vocês devem calcular o ranking de cada mês baseado nos jogos até o mês anterior. Como um ranking depende do ranking anterior, disponibilizaremos o ranking de Julho 2006, quando a forma atual de cálculo foi adotada. Você deve montar o ranking mês a mês a partir do ranking de Agosto 2006 até Maio de 2018. Apenas o ranking de Maio 2018 deve ser incluído no arquivo de saída. O formato deve ser (os dados abaixo são reais de Maio 2018):

```
Posicao,Pontuacao,Pais
1,1544,Germany
2,1384,Brazil
3,1346,Belgium
...
```

Leia com atenção o link acima para que possa fazer os cálculos corretamente. O arquivo com os jogos não inclui disputas de pênaltis (o vencedor recebe 2 pontos), logo não será possível que o ranking de vocês seja exatamente igual ao ranking oficial. A aproximação porém deve ser boa o suficiente.

O que deve ser feito

1. Implementar um programa em linguagem C que lê o arquivo de entrada e escreve o arquivo de saída. O arquivo de saída pode ter o nome fixo `saida.txt`. Se fizer a tarefa extra, crie um novo arquivo de saída com o nome `ranking.txt`.
2. Documentação incluindo uma descrição do programa, incluindo os algoritmos implementados, análise de complexidade a uma tabela comparando o tempo de execução com cada método de busca diferente. Se considerar relevante, outras informações podem ser incluídas.

O que deve ser entregue

- Código fonte do programa em C (todos os arquivos `.c` e `.h`), bem identada e comentada, conforme especificado acima - não inclua o executável.
- Documentação sobre o trabalho (em `.pdf`)
- Não inclua o executável, apenas os arquivos com código fonte.
- Já falei para não incluir o executável?

Submissão

A forma de submissão do trabalho será via Moodle. Ao receber seu código, nós compilaremos seu programa em um ambiente Linux com o seguinte comando:

```
gcc -Wall -std=c99 *.c -o tp2
```

Não há problema algum em desenvolver o trabalho no Windows, apenas tome cuidado para não utilizar bibliotecas que não compilam no Linux (por exemplo: `<windows.h>`). Nos próximos dias mostrarei para vocês em sala funções de medição de tempo em C que podem ser utilizadas. Também incluirei um arquivo de exemplo no Moodle.

Comentários gerais

1. Comece a fazer este trabalho logo, enquanto o problema está fresco na memória e o prazo para terminá-lo está tão longe quanto jamais poderá estar.
2. Clareza, indentação e comentários no programa também serão avaliados.
3. Uma dica: faça uma parte de cada vez. Teste bem e só passe para a próxima etapa quando tiver certeza que o que fez está funcionando corretamente.
4. O trabalho é individual.
5. Trabalhos copiados, comprados, doados, etc. serão penalizados severamente.
6. Penalização por atraso: $(2^d - 1)$ pontos, onde d é o número de dias de atraso.