

Universidade Federal de Ouro Preto - UFOP
Departamento de Ciência da Computação - DECOM

Relatório atividade 4 - Football(akasoccer)

BCC402 - ALGORITMOS E PROGRAMACAO AVANCADA

Kayo Xavier Nascimento Cavalcante Leite - 21.2.4095

Professor: Rafael Alves Bonfim

Ouro Preto
31 de março de 2025

Sumário

1	Código e enunciado.	1
2	Pseudocódigo e descrição do problema	1
3	Problema 10194: Football (aka Soccer)	1
3.1	Regras de Classificação	1
3.2	Entrada e Saída	1
3.3	Estratégia de Solução	2
3.4	Detalhes de Implementação	2
4	Casos teste - Input e output esperado.	3

Lista de Códigos Fonte

1	Pseudocódigo do problema.	2
---	-----------------------------------	---

1 Código e enunciado.

Na Atividade 4 - o problema selecionado foi Football(aka soccer). O Problema tem como objetivo calcular os resultados finais e tabelas de campeonatos através do input das partidas e resultados de jogos seguindo critérios específicos de desempate. O código comentado e documentado, casos de teste e executável pré compilado se encontram no .zip da atividade.

Caso queira, para rodar e compilar o código, é necessário ter o compiler g++ e utilizar o seguinte comando no terminal dentro do diretório da pasta da atividade específica:

Compilando e rodando o exercício

```
para compilar:
g++ Football(akasoccer).cpp -o executavel

e para rodar basta utilizar .\executavel no cmd.

para utilizar os cenários de teste:
.\executavel < sampleinput.txt
.\executavel < testinput.txt
```

2 Pseudocódigo e descrição do problema

3 Problema 10194: Football (aka Soccer)

Objetivo: Calcular a classificação de times em um torneio com base em jogos disputados, seguindo critérios específicos de desempate.

3.1 Regras de Classificação

Os times são ordenados por:

1. Maior número de pontos (3 por vitória, 1 por empate)
2. Maior número de vitórias
3. Maior saldo de gols (gols marcados - gols sofridos)
4. Maior número de gols marcados
5. Menor número de jogos disputados
6. Ordem lexicográfica (case insensitive)

3.2 Entrada e Saída

- **Entrada:**

- Número de torneios N
- Para cada torneio:
 - * Nome do torneio
 - * Número de times T e seus nomes
 - * Número de jogos G e resultados no formato `time1#gols1@gols2#time2`

- **Saída:**

- Para cada torneio:
 - * Nome do torneio

* Classificação no formato:
 [posição] [time] [pontos]p, [jogos]g ([vitórias]-[empates]-[derrotas]),
 [saldo]gd ([gols pró]-[gols contra])

3.3 Estratégia de Solução

Estruturas-Chave:

- **TeamStats**: Armazena estatísticas de cada time (pontos, vitórias, gols, etc.)
- **map<string, TeamStats>**: Mapeia nomes de times para suas estatísticas

Passos:

1. Leitura dos Dados:

- Para cada torneio, ler nomes dos times e pré-inicializar estatísticas.
- Processar cada jogo:
 - Extrair gols e times usando operações de substring
 - Atualizar estatísticas (pontos, vitórias, gols, etc.)

2. Ordenação:

- Converter o mapa para um vetor de pares (**nome**, **estatísticas**)
- Ordenar usando função **CompareTeams** que implementa as regras de desempate

3. Impressão:

- Formatar saída conforme especificado usando **printf**

3.4 Detalhes de Implementação

- **Parsing de Jogos**:

Exemplo: "TeamA#3@1#TeamB" → TeamA marcou 3, TeamB marcou 1

- **Atualização de Estatísticas**:

Pontos = $3 \times \text{Vitórias} + 1 \times \text{Empates}$

Saldo = Gols Marcados – Gols Sofridos

- **Ordenação Case-Insensitive**:

- Armazenar versão em lowercase do nome (**lower_name**)
- Comparar usando **lower_name** na função de ordenação

```

1
2 // Classificacao de Torneios
3
4 Estrutura TimeStats:
5   pontos, jogos, vitorias, empates, derrotas, gols_pro, gols_contra,
6   nome_lower
7
8 Fun    o CompararTimes(a, b): // Regras de ordena    o
9   1. Pontos (maior)
10  2. Vit rias (maior)
11  3. Saldo de Gols (maior)
12  4. Gols Pr   (maior)
13  5. Jogos (menor)

```

```

13 6. Nome (alfabética, case-insensitive)
14 Retorna true se 'a' precede 'b'
15
16 // Principal
17 Ler num_torneios
18 primeiro_torneio = true
19
20 Enquanto num_torneios > 0:
21   Se não primeiro_torneio, Imprimir linha_vazia
22   primeiro_torneio = false
23
24   Ler nome_torneio
25   Imprimir nome_torneio
26
27   Ler num_times
28   Mapa times_stats (nome -> TimeStats) // Armazena estatísticas
29
30   // Ler nomes e inicializar times no mapa
31   Para i de 1 at num_times:
32     Ler nome_time
33     Inicializar times_stats[nome_time] (stats zeradas, nome_lower calculado)
34
35   Ler num_jogos
36   // Ler e processar resultados dos jogos
37   Para i de 1 at num_jogos:
38     Ler linha_jogo ("TimeA#GolsA@GolsB#TimeB")
39     Extrair timeA, golsA, golsB, timeB
40     Atualizar estatísticas (jogos, gols, pontos, V/E/D) para timeA e timeB em
        times_stats
41
42   // Ordenar e imprimir
43   Criar Vetor times_ordenados a partir de times_stats
44   Ordenar times_ordenados usando CompararTimes
45
46   Imprimir cabeçalho da classificação (se necessário)
47   rank = 1
48   Para cada time em times_ordenados:
49     Imprimir (rank, nome, pontos, jogos, V-E-D, saldo, gols_pro-gols_contra)
50     rank++
51
52   num_torneios-- // Decrementa contador

```

Código 1: Pseudocódigo do problema.

4 Casos teste - Input e output esperado.

Para os casos de teste do problema, foi disponibilizado junto a pasta do mesmo os seguintes arquivos :sampleinput.txt e testinput.txt, sendo o primeiro o próprio caso de teste disponibilizado pelo exercício e o segundo caso de teste encontrado na plataforma <https://www.udebug.com/>. Além disso, encontra-se também o arquivo com os outputs esperados para cada input. Ambos os resultados foram validados e tiveram o output esperado.

sampleinput.txt

```
2
World Cup 1998 - Group A
4
Brazil
Norway
Morocco
Scotland
6
Brazil#2@1#Scotland
Norway#2@2#Morocco
Scotland#1@1#Norway
Brazil#3@0#Morocco
Morocco#3@0#Scotland
Brazil#1@2#Norway
Some strange tournament
5
Team A
Team B
Team C
Team D
Team E
5
Team A#1@1#Team B
Team A#2@2#Team C
Team A#0@0#Team D
Team E#2@1#Team C
Team E#1@2#Team D
```

output esperado

```
World Cup 1998 - Group A
1) Brazil 6p, 3g (2-0-1), 3gd (6-3)
2) Norway 5p, 3g (1-2-0), 1gd (5-4)
3) Morocco 4p, 3g (1-1-1), 0gd (5-5)
4) Scotland 1p, 3g (0-1-2), -4gd (2-6)
Some strange tournament
1) Team D 4p, 2g (1-1-0), 1gd (2-1)
2) Team E 3p, 2g (1-0-1), 0gd (3-3)
3) Team A 3p, 3g (0-3-0), 0gd (3-3)
4) Team B 1p, 1g (0-1-0), 0gd (1-1)
5) Team C 1p, 2g (0-1-1), -1gd (3-4)
```

Os demais testes se encontram no diretório da atividade