

Aufgabe 3: **Tobis Turnier**

Team-ID: **00255**

Team: **Solo**

Bearbeiter dieser Aufgabe: **Walther
Trgovac**

20. November 2020

Inhaltsverzeichnis

Lösungsidee.....	1
Umsetzung.....	1
Beispiele.....	3
Quellcode.....	11

Lösungsidee

Das Ziel der Aufgabe ist es, einfach diese Turnierformen zu simulieren und dann gucken, bei welcher Variante der Spieler mit der größten Spielstärke am meisten gewinnt. Es gibt 3 Turnierformen: Liga, K.O., K.O. x5. Ich habe noch 2 implementiert und das sind dieselben Formen wie K.O. und K.O. x5, aber der Unterschied ist nur, dass die Positionen der Spieler zufällig ausgesucht sind.

Jede Form hat ihre Vorteile und Nachteile. Bei K.O. Ist der Nachteil, dass nicht jeder Spieler gegen jeden spielt und so kann man den stärkeren Spieler manchmal vermeiden und gegen ihn überhaupt nicht spielen. Bei Liga ist der Nachteil, dass es manchmal langweilig ist. Wenn der stärkste Spieler von Anfang an dominiert, ist das Turnier noch nicht zu Ende, aber man weiß es schon, wer gewinnen wird. Aber meiner Meinung nach ist Liga sehr gut, weil jeder gegen jeden einmal spielt und das ist ihr größter Vorteil.

Weil es hier um Wahrscheinlichkeit geht, ist es am besten, wenn so Spiele gespielt werden wie möglich. Das bedeutet, dass K.O. x5 auf jeden Fall besser ist als K.O..

Welche Form am besten ist, werden wir bei den Beispielen sehen.

Umsetzung

Meine Lösungsidee habe ich in C++ implementiert.

Liga Form zu simulieren ist sehr einfach. Jeder muss gegen jeden einmal spielen und das kann man sehr einfach mit Hilfe des double Loops machen. Man muss nur aufpassen, falls 2 Spieler dieselbe Anzahl der Siege haben, dass der Spieler mit der kleineren Spielnummer gewinnt. Das Problem habe ich so gelöst, indem ich einen Vektor habe, in dem ich nicht nur die Anzahl der Siege und die Spielstärke speichere, sondern auch die Spielnummer. Dann am Ende wird die Liste mit den Gewinnern sortiert, sodass der Spieler mit der kleineren Spielnummer am Anfang steht.

K.O. ist bisschen schwieriger, aber auch machbar. Bei jeder Runde wird die Anzahl der Spieler doppelt so klein wie in der vorherigen Runde. Zum Beispiel: $8 \rightarrow 4 \rightarrow 2 \rightarrow 1$. Wenn man das addiert, kommt man auf die Summe von 15. Wenn $8 \cdot n$ ist, dann ist diese Summe $2 \cdot n - 1$. Das ist sehr hilfreich, weil man eine Liste machen kann und dann alle Spieler in diese Liste speichern kann. Dann wenn Einer gewinnt, bekommt er einen Platz weiter in der Liste und so macht man das weiter, bis nur Einer übrig geblieben ist. Der Spieler an der letzten Position in der Liste gewinnt.

Zum Beispiel:

(1, 2, 3, 4, 5, 6, 7, 8, 1, 3, 5, 7, 1, 5, 1)

1. Runde
2. Runde
3. Runde
4. Runde

Hier sieht man eine Liste mit Spielern, die von 1 bis 8 nummeriert sind. Dann spielt 1 gegen 2 und 1 gewinnt. Das bedeutet, er bekommt den nächsten Platz in der Liste, dann spielt 3 gegen 4 und 3 gewinnt. Er bekommt den nächsten Platz und so weiter, bis die Liste voll ist. Der Spieler am Ende, in diesem Fall Spieler 1, gewinnt.

Der einzige Unterschied zwischen K.O. normal und K.O. mit zufälligen Positionen, ist das bei K.O. mit zufälligen Positionen sind sie nicht so positioniert wie bei der Eingabe, sondern sind die zufällig generiert mit Hilfe der Funktion "random_shuffle".

Wie der Gewinner entschieden wird, ist genauso wie in der Aufgabenstellung beschrieben.

Mein Programm durchführt die Turnierformen nicht nur einmal, sondern 5000 Mal, damit das Ergebnis präziser ist. Ich habe das so gemacht, indem ich eine Liste mit allen Spielern gemacht habe und wenn Einer gewinnt, wird einfach die Anzahl der Siege dieses Spielers addiert.

Meine Lösung funktioniert für jede Anzahl der Spieler, die eine Zweierpotenz ist.

Beispiele

Im Beispiel 1 werde ich die komplette Ausgabe des Programms zeigen, aber in den nächsten nur die 3 besten Spieler, damit es nicht so lang ist.

Beispiel 1:

Eingabe:

8
0
10
20
30
40
50
60
100

Ausgabe:

Liga Variante (5000 Spiele):

Spieler 1 hat 0 Mal gewonnen. (0%)
Spieler 2 hat 35 Mal gewonnen. (0%)
Spieler 3 hat 199 Mal gewonnen. (3%)
Spieler 4 hat 490 Mal gewonnen. (9%)
Spieler 5 hat 723 Mal gewonnen. (14%)
Spieler 6 hat 860 Mal gewonnen. (17%)
Spieler 7 hat 1037 Mal gewonnen. (20%)
Spieler 8 hat 1656 Mal gewonnen. (33%)

K.O. (normal) Variante (5000 Spiele):

Spieler 1 hat 0 Mal gewonnen. (0%)
Spieler 2 hat 198 Mal gewonnen. (3%)
Spieler 3 hat 315 Mal gewonnen. (6%)
Spieler 4 hat 693 Mal gewonnen. (13%)
Spieler 5 hat 487 Mal gewonnen. (9%)
Spieler 6 hat 759 Mal gewonnen. (15%)
Spieler 7 hat 762 Mal gewonnen. (15%)
Spieler 8 hat 1786 Mal gewonnen. (35%)

K.O. (zufällige Positionen) Variante (5000 Spiele):

Spieler 1 hat 0 Mal gewonnen. (0%)
Spieler 2 hat 35 Mal gewonnen. (0%)
Spieler 3 hat 179 Mal gewonnen. (3%)
Spieler 4 hat 341 Mal gewonnen. (6%)
Spieler 5 hat 561 Mal gewonnen. (11%)
Spieler 6 hat 819 Mal gewonnen. (16%)
Spieler 7 hat 1023 Mal gewonnen. (20%)
Spieler 8 hat 2042 Mal gewonnen. (40%)

K.O. 5 (normale Positionen) Variante (5000 Spiele):

Spieler 1 hat 0 Mal gewonnen. (0%)
Spieler 2 hat 3 Mal gewonnen. (0%)
Spieler 3 hat 97 Mal gewonnen. (1%)
Spieler 4 hat 487 Mal gewonnen. (9%)
Spieler 5 hat 301 Mal gewonnen. (6%)
Spieler 6 hat 613 Mal gewonnen. (12%)
Spieler 7 hat 699 Mal gewonnen. (13%)
Spieler 8 hat 2800 Mal gewonnen. (56%)

K.O. 5 (zufällige Positionen) Variante (5000 Spiele):

Spieler 1 hat 0 Mal gewonnen. (0%)
Spieler 2 hat 0 Mal gewonnen. (0%)
Spieler 3 hat 26 Mal gewonnen. (0%)
Spieler 4 hat 124 Mal gewonnen. (2%)
Spieler 5 hat 291 Mal gewonnen. (5%)
Spieler 6 hat 578 Mal gewonnen. (11%)
Spieler 7 hat 980 Mal gewonnen. (19%)
Spieler 8 hat 3001 Mal gewonnen. (60%)

Hier sieht man, dass bei K.O. und K.O. 5 der Spieler mit der größten Spielstärke die größte Anzahl der Siege hat.

Beispiel 2:

Eingabe:

8

10101010

808080100

Ausgabe:

Liga Variante (5000 Spiele):

Spieler 5 hat 1701 Mal gewonnen. (34%)
Spieler 6 hat 1212 Mal gewonnen. (24%)
Spieler 8 hat 1052 Mal gewonnen. (21%)

K.O. (norMal) Variante (5000 Spiele):

Spieler 5 hat 1047 Mal gewonnen. (20%)
Spieler 6 hat 1028 Mal gewonnen. (20%)
Spieler 8 hat 1416 Mal gewonnen. (28%)

K.O. (zufällige Positionen) Variante (5000 Spiele):

Spieler 5 hat 1129 Mal gewonnen. (22%)
Spieler 6 hat 1176 Mal gewonnen. (23%)
Spieler 8 hat 1477 Mal gewonnen. (29%)

K.O. 5 (norMale Positionen) Variante (5000 Spiele):

Spieler 5 hat 1085 Mal gewonnen. (21%)
Spieler 6 hat 1095 Mal gewonnen. (21%)
Spieler 8 hat 1823 Mal gewonnen. (36%)

K.O. 5 (zufällige Positionen) Variante (5000 Spiele):

Spieler 6 hat 1079 Mal gewonnen. (21%)

Spieler 7 hat 1056 Mal gewonnen. (21%)

Spieler 8 hat 1809 Mal gewonnen. (36%)

In diesem Beispiel gewinnt die K.O. 5 Form schon wieder, hier sieht man deutlich, je mehr Spiele spielt man, desto besser das ist für den Spieler mit der größten Spielstärke.

Beispiel 3:

Eingabe:

16

22386693515158675157576073134142

Ausgabe:

Liga Variante (5000 Spiele):

Spieler 3 hat 769 Mal gewonnen. (15%)

Spieler 4 hat 1633 Mal gewonnen. (32%)

Spieler 8 hat 455 Mal gewonnen. (9%)

K.O. (normal) Variante (5000 Spiele):

Spieler 3 hat 432 Mal gewonnen. (8%)

Spieler 4 hat 817 Mal gewonnen. (16%)

Spieler 13 hat 781 Mal gewonnen. (15%)

K.O. (zufällige Positionen) Variante (5000 Spiele):

Spieler 4 hat 838 Mal gewonnen. (16%)

Spieler 8 hat 435 Mal gewonnen. (8%)

Spieler 13 hat 523 Mal gewonnen. (10%)

K.O. 5 (normale Positionen) Variante (5000 Spiele):

Spieler 3 hat 417 Mal gewonnen. (8%)

Spieler 4 hat 1416 Mal gewonnen. (28%)

Spieler 13 hat 1120 Mal gewonnen. (22%)

K.O. 5 (zufällige Positionen) Variante (5000 Spiele):

Spieler 4 hat 1368 Mal gewonnen. (27%)

Spieler 8 hat 534 Mal gewonnen. (10%)

Spieler 13 hat 692 Mal gewonnen. (13%)

In diesem Fall gewinnt der Spieler mit der größten Spielstärke jedes Mal, aber in diesem Fall ist der Spieler mit der größten Anzahl der Siege in der Liga Form.

Beispiel 4:

Eingabe:

16
100
95
95
95
95
95
95
95
95
95
95
95
95
95
95
95

Ausgabe:

Liga Variante (5000 Spiele):

Spieler 1 hat 569 Mal gewonnen. (11%)
Spieler 2 hat 444 Mal gewonnen. (8%)
Spieler 3 hat 360 Mal gewonnen. (7%)
Spieler 4 hat 396 Mal gewonnen. (7%)
Spieler 5 hat 366 Mal gewonnen. (7%)
Spieler 6 hat 349 Mal gewonnen. (6%)
Spieler 7 hat 339 Mal gewonnen. (6%)
Spieler 8 hat 278 Mal gewonnen. (5%)
Spieler 9 hat 279 Mal gewonnen. (5%)
Spieler 10 hat 255 Mal gewonnen. (5%)
Spieler 11 hat 256 Mal gewonnen. (5%)
Spieler 12 hat 254 Mal gewonnen. (5%)
Spieler 13 hat 211 Mal gewonnen. (4%)
Spieler 14 hat 206 Mal gewonnen. (4%)
Spieler 15 hat 227 Mal gewonnen. (4%)
Spieler 16 hat 211 Mal gewonnen. (4%)

K.O. (normal) Variante (5000 Spiele):

Spieler 1 hat 336 Mal gewonnen. (6%)
Spieler 2 hat 288 Mal gewonnen. (5%)
Spieler 3 hat 290 Mal gewonnen. (5%)
Spieler 4 hat 331 Mal gewonnen. (6%)
Spieler 5 hat 332 Mal gewonnen. (6%)

Spieler 6 hat 296 Mal gewonnen. (5%)
Spieler 7 hat 300 Mal gewonnen. (6%)
Spieler 8 hat 312 Mal gewonnen. (6%)
Spieler 9 hat 320 Mal gewonnen. (6%)
Spieler 10 hat 294 Mal gewonnen. (5%)
Spieler 11 hat 314 Mal gewonnen. (6%)
Spieler 12 hat 317 Mal gewonnen. (6%)
Spieler 13 hat 347 Mal gewonnen. (6%)
Spieler 14 hat 295 Mal gewonnen. (5%)
Spieler 15 hat 320 Mal gewonnen. (6%)
Spieler 16 hat 308 Mal gewonnen. (6%)

K.O. (zufällige Positionen) Variante (5000 Spiele):

Spieler 1 hat 334 Mal gewonnen. (6%)
Spieler 2 hat 266 Mal gewonnen. (5%)
Spieler 3 hat 325 Mal gewonnen. (6%)
Spieler 4 hat 298 Mal gewonnen. (5%)
Spieler 5 hat 329 Mal gewonnen. (6%)
Spieler 6 hat 342 Mal gewonnen. (6%)
Spieler 7 hat 315 Mal gewonnen. (6%)
Spieler 8 hat 279 Mal gewonnen. (5%)
Spieler 9 hat 343 Mal gewonnen. (6%)
Spieler 10 hat 330 Mal gewonnen. (6%)
Spieler 11 hat 324 Mal gewonnen. (6%)
Spieler 12 hat 300 Mal gewonnen. (6%)
Spieler 13 hat 282 Mal gewonnen. (5%)
Spieler 14 hat 302 Mal gewonnen. (6%)
Spieler 15 hat 304 Mal gewonnen. (6%)
Spieler 16 hat 327 Mal gewonnen. (6%)

K.O. 5 (normale Positionen) Variante (5000 Spiele):

Spieler 1 hat 392 Mal gewonnen. (7%)
Spieler 2 hat 286 Mal gewonnen. (5%)
Spieler 3 hat 308 Mal gewonnen. (6%)
Spieler 4 hat 270 Mal gewonnen. (5%)
Spieler 5 hat 293 Mal gewonnen. (5%)
Spieler 6 hat 308 Mal gewonnen. (6%)
Spieler 7 hat 300 Mal gewonnen. (6%)
Spieler 8 hat 301 Mal gewonnen. (6%)
Spieler 9 hat 298 Mal gewonnen. (5%)
Spieler 10 hat 350 Mal gewonnen. (7%)
Spieler 11 hat 291 Mal gewonnen. (5%)
Spieler 12 hat 343 Mal gewonnen. (6%)
Spieler 13 hat 318 Mal gewonnen. (6%)
Spieler 14 hat 324 Mal gewonnen. (6%)
Spieler 15 hat 308 Mal gewonnen. (6%)

Spieler 16 hat 310 Mal gewonnen. (6%)

K.O. 5 (zufällige Positionen) Variante (5000 Spiele):

Spieler 1 hat 399 Mal gewonnen. (7%)

Spieler 2 hat 293 Mal gewonnen. (5%)

Spieler 3 hat 320 Mal gewonnen. (6%)

Spieler 4 hat 300 Mal gewonnen. (6%)

Spieler 5 hat 306 Mal gewonnen. (6%)

Spieler 6 hat 307 Mal gewonnen. (6%)

Spieler 7 hat 336 Mal gewonnen. (6%)

Spieler 8 hat 327 Mal gewonnen. (6%)

Spieler 9 hat 288 Mal gewonnen. (5%)

Spieler 10 hat 295 Mal gewonnen. (5%)

Spieler 11 hat 313 Mal gewonnen. (6%)

Spieler 12 hat 283 Mal gewonnen. (5%)

Spieler 13 hat 310 Mal gewonnen. (6%)

Spieler 14 hat 318 Mal gewonnen. (6%)

Spieler 15 hat 292 Mal gewonnen. (5%)

Spieler 16 hat 313 Mal gewonnen. (6%)

Hier ist es sehr knapp, deswegen habe ich nicht nur die drei besten gelassen, sodass man es sieht. Aber trotzdem hat der Spieler 1 in der Liga Form die größte Anzahl der Siege.

Ich habe noch ein Beispiel, mit 32 Spielern. Die erste Hälfte der Spieler ist gleich wie bei Beispiel 3, nur noch 16 neue Spieler sind addiert.

Beispiel 5:

Eingabe:

32
22
38
66
93
51
51
58
67
51
57
57
60
73
13
41
42
80
53
21
43
56
64
12
34
56
74
14
39
60
30
20
51

Ausgabe:

Liga Variante (5000 Spiele):

Spieler 4 hat 1505 Mal gewonnen. (30%)

Spieler 17 hat 608 Mal gewonnen. (12%)

Spieler 26 hat 352 Mal gewonnen. (7%)

K.O. (normal) Variante (5000 Spiele):

Spieler 3 hat 230 Mal gewonnen. (4%)

Spieler 13 hat 423 Mal gewonnen. (8%)

Spieler 17 hat 433 Mal gewonnen. (8%)

K.O. (zufällige Positionen) Variante (5000 Spiele):

Spieler 4 hat 561 Mal gewonnen. (11%)

Spieler 17 hat 417 Mal gewonnen. (8%)

Spieler 26 hat 336 Mal gewonnen. (6%)

K.O. 5 (normale Positionen) Variante (5000 Spiele):

Spieler 4 hat 978 Mal gewonnen. (19%)

Spieler 13 hat 622 Mal gewonnen. (12%)

Spieler 17 hat 644 Mal gewonnen. (12%)

K.O. 5 (zufällige Positionen) Variante (5000 Spiele):

Spieler 4 hat 967 Mal gewonnen. (19%)

Spieler 17 hat 595 Mal gewonnen. (11%)

Spieler 26 hat 465 Mal gewonnen. (9%)

Diesmal gibt es 32 Spieler und Spieler 4 gewinnt 1505 Mal. An diesem Beispiel sieht man deutlich, dass die Liga Form die beste Form ist. Das ist auch logisch und fair, weil wie oben schon geschrieben, jeder spielt gegen jeden einmal und so ist es am fairsten. Deswegen würde ich mich für die **Liga Form** entscheiden.

Quellcode

```
//Liga
//Jeder spielt gegen jeden einmal, falls er gewinnt, erhöht sich seine Anzahl
der Siege um 1
for (int Spiele = 0; Spiele < 5000; Spiele++){
    vector <pair <int, pair <int,int>>> liga(n);
    for (int i = 0; i < n; i++){
        liga[i].first = liga2[i].first;
        liga[i].second.first = liga2[i].second.first;
        liga[i].second.second = liga2[i].second.second; }

for (int i = 0; i < n; i++){
    for (int j = i+1; j < n; j++){
        // zufaellige Zahl wird generiert
        double r1 = mt_rand() / (double)mt_rand.max();
        // beide Spieler werden definiert
        double player1 = liga[i].first;
        double player2 = liga[j].first;
        double sum = player1+player2;
        // Der selbe Prozess wie schon in der Aufgabenstellung beschrieben ist
        if (r1 <= player1/sum){
            liga[i].second.first++; }
        else {
            liga[j].second.first++;
        }
    }
}
```

Hier ist K.O.5 mit normalen Positionen, der einzige Unterschied zwischen K.O.5 und K.O. normal ist der for Loop (j). Bei K.O. normal gibt es diesen Loop nicht, da jeder gegen jeden nur einmal spielt.

```
//KO5normal
for (int Spiele = 0; Spiele < 5000; Spiele++){
    int pos = 0;
    for (int i = n; i < 2*n-1; i++){
        double player1 = ko5[pos].first;
        double player2 = ko5[pos+1].first;
        double sum = player1+player2;
        int cnt1 = 0, cnt2 = 0;
        // In dieser For Loop wird entscheidet, wer gewinnt und in die naechste
        Runde kommt
        for (int j = 0; j < 5; j++){
            double r4 = mt_rand() / (double)mt_rand.max();
            if (r4 <= player1/sum){
                cnt1++; }
            else {
                cnt2++;
            }
        }
    }
}
```

```

    if (cnt1 > cnt2){
        ko5[i].first=ko5[pos].first;
        ko5[i].second=ko5[pos].second;
    } else {
        ko5[i].first=ko5[pos+1].first;
        ko5[i].second=ko5[pos+1].second;
    }
    pos+=2;
}
AlleGewinner[ko5[2*n-2].second-1]++;
}

```

Der Unterschied bei K.O. mit zufällig generierten Positionen und normalen Positionen ist, dass bei zufällig generierten Funktion "random_shuffle" verwendet wird, sonst ist alles genauso wie oben.