

Tema 1: LAN Party

Obiective

- Înțelegerea conceptului de funcționare și implementare a unor structuri de date (precum liste, stive, cozi, arbori binari de căutare)
- Operarea cu aceste structuri de date
- Implementarea unei funcționalități practice folosind aceste concepte

Informații

- Deadline hard, 6.12.2020 ora 23:59 (termen limită - nu se obțin puncte pe soluțiile trimise mai târziu)
- Checker-ul offline îl puteți descărca de la această adresă [<https://github.com/poo-is/lan-party-checker>]
- Puteți cere ajutor oricând la această adresă email [<mailto:poo.is.teme@gmail.com>]

Descriere

Facultatea de Automatica și Calculatoare prin LSAC organizează în fiecare semestru un LAN Party la care participă toți studenții care vor să își testeze abilitățile la un anumit joc. Se dorește realizarea unei soluții software (program executabil) ce poate automatiza acest proces.

Cerințe

1. Se va crea o listă cu informațiile din fișierul d.in (lista se va forma prin adăugarea elementelor la început - $O(1)$).
2. Se vor elimina din lista anterior creată, echipele cu cel mai mic punctaj de echipă (informațiile despre modul de calcul al acestui punctaj se găsesc în cadrul secțiunii Detalii Tehnice) astfel încât în lista să rămână n echipe (unde n maxim și n este puterea lui 2). În cazul în care mai multe echipe au același scor se elimină prima echipă găsită în lista. După fiecare eliminare se va relua parcurgerea listei de la cap.
3. Se vor implementa structurile de date necesare pentru a se efectua meciurile:
 - Se va crea o coadă în care se vor pune meciurile dintre echipe.
 - Echipele vor forma meciuri în ordinea în care se află în lista de la punctul anterior, primele două echipe vor forma primul meci (și așa mai departe).
 - Se vor crea două stive - una pentru învinși și una pentru câștigători - ce vor conține echipele ce au pierdut, respectiv, au câștigat meciul anterior.
 - Se vor introduce în stivă echipele în funcție de rezultatul obținut și se va adăuga un punct la punctajul de echipă al echipelor ce au câștigat.
 - În cazul unui meci în care ambele echipe au același punctaj, se va alege prima echipă adăugată în meciul disputat.
 - Se vor șterge echipele ce se găsesc în stiva de învinși.
 - Din stiva de câștigători se vor extrage echipele și se va umple din nou coada de meciuri formându-se meciuri cu echipe consecutive (prima echipă scoasă din stivă va juca cu a doua echipă scoasă din stivă primul meci).
 - Se vor repeta pașii enumerați până la aflarea echipelor de pe primele 8 locuri, care vor fi salvate într-o altă listă, după aflarea acestora se vor continua meciurile până la aflarea echipei castigatoare.
4. Se va realiza un clasament al ultimelor 8 echipe (salvate la subpunctul anterior) sub forma unui BST (Binary Search Tree). Acesta va fi ordonat după punctajul pe echipă și i se va afișa conținutul în ordine descrescătoare (în cazul în care două echipe au același punctaj se vor ordona în funcție de nume, descrescător).

5. Se va realiza un arbore de tip AVL cu ultimele 8 echipe (salvate la subpunctul 3, dar sortate ca la subpunctul 4) si se vor afisa echipele de pe nivelul 2 al arborelui.

Punctaje si Detalii Tehnice

Punctaje:

- Cerintele 1-3: 70% nota
- Cerinta 4: 20% nota
- Cerinta 5: 10% nota

Detalii Tehnice

- Rezultatele obținute în urma executării cerințelor din fișierul c.in vor fi scrise în fișierul r.out
- Pentru cerințele 1 si 2 este suficienta scrierea în fișierul de rezultate a numelor echipelor, în ordinea din lista creata, pe cate o linie separata.
- Pentru cerința 3 se va scrie în fișier la fiecare runda conținutul cozii de meciuri alături de conținutul stivei de învingători.
- Pentru cerința 4 se dorește afișarea BST -ului în ordine descrescătoare. Adică afișarea clasamentului în ordine descrescătoare.
- Pentru cerința 5 se dorește crearea unui AVL cu echipele de la sub punctul anterior și scrierea în fișier a echipelor de la nivelul 2.
- Structurile de date necesare trebuie deduse astfel încât sa respecte normele de buna implementare Ex: un jucator are nume, prenume, experienta - prin urmare se va folosi un tip de date care sa caracterizeze aceasta entitate:(Acelasi lucru trebuie facut si pentru alte tipuri de date necesare realizarii aplicatiei.)

player.hpp

```
class Player
{
private:
    string firstName;
    string secondName;
    int points;
};
```

Formula de calcul a punctajului de echipa este: media aritmetica a experienței (xp) jucătorilor din echipa respectiva. La fiecare meci câștigat punctajul de echipa va crește cu 1.(Trebuie modificate toate punctajele jucătorilor echipei pentru realizarea acestui task.)

Exemplu

Fisierele folosite de checker arata astfel:

c.in

```
1 0 0 0 0
```

Obiectivele se vor realiza în funcție de cum apar în fișierul cu cerinte cerinte.in (Spre exemplu dacă fișierul cerinte.in conține 1 0 0 0 0 înseamnă ca se dorește doar realizarea cerinței 1. Se vor folosi ca valori de intrare datele din fișierul date.in)

d.in

```
45 // numarul de echipe din fisier
10 The Waiver Wire // numarul de jucatori si numele echipei
Popa Ana-Maria 6 // nume prenume puncte jucator 1
Popescu Alexandru 10
```

Pop Mihaela 6
Radu Andreea 2
Dumitru Elena 1
Stan Adrian 4
Stoica Andrei 0
Gheorghe Alexandra 6
Matei Mihai 3
Ciobanu Ionut 1

10 The Big Egos
Ionescu Grigore 8
Rusu Cosmin-Alexandru 7
Mihai Ionut-Florin 5
Serban Victoria 3
Constantin Marius-Valentin 7
Marin Elena-Claudia 4
Stefan Maria-Roxana 9
Lazar Alina-Daniela 10
Vasile Andreea-Simona 2
Munteanu Cristina-Daniela 0

10 Chafing the Dream
Ilie Cristian-Mihai 10
Oprea Mihai-Catalin 8
Toma Andra-Maria 5
Florea Ioana-Mihaela 0
Moldovan Alexandru-Iulian 4
Tudor Bogdan-Ioan 6
Stanci Alexandra-Georgiana 0
Sandu Cristina-Gabriela 10
Ion Andreea-Catalina 3
Andrei Marius-Mihai 10

r.out

VALLEY GIRLS
TWISTING TORNADOES
TOUCANS
TIMBERWOLVES
TERMINATORS
TEAL TITANS
SWEETHEARTS
SUPER POWER GIRLS
SUGER BEES
STINGRAYS
STAR SHOOTERS
SPEEDY TURTLES
SPACE BABES
SOCCER DIVAS
SNOW DEVILS
SLICK CHICS
SILVER STRIKE
SILVER BUTTERFLIES
SHARKS
SCORPIONS
MUTINY
MIND TRIP
MIGHTY PANTHERS
MIDNIGHT RAIDERS
MEAN GREEN ALIENS
MANARCHS
FUGITIVES
FLASH
FLAMING TIGERS
FIRE STRIKERS
FIGHTING IRISH
AVALANCHE

--- ROUND NO:1
VALLEY GIRLS
TOUCANS
TERMINATORS
SWEETHEARTS
SUGER BEES
STAR SHOOTERS
SPACE BABES
SNOW DEVILS
SILVER STRIKE
SHARKS

TWISTING TORNADOES
TIMBERWOLVES
TEAL TITANS
SUPER POWER GIRLS
STINGRAYS
SPEEDY TURTLES
SOCCER DIVAS
SLICK CHICS
SILVER BUTTERFLIES
SCORPIONS

MUTINY	-	MIND TRIP
MIGHTY PANTHERS	-	MIDNIGHT RAIDERS
MEAN GREEN ALIENS	-	MANARCHS
FUGITIVES	-	FLASH
FLAMING TIGERS	-	FIRE STRIKERS
FIGHTING IRISH	-	AVALANCHE
WINNERS OF ROUND NO:1		
FIGHTING IRISH	-	7.80
FLAMING TIGERS	-	6.20
FUGITIVES	-	7.00
MEAN GREEN ALIENS	-	6.60
MIDNIGHT RAIDERS	-	5.70
MIND TRIP	-	6.90
SCORPIONS	-	6.80
SILVER BUTTERFLIES	-	7.00
SLICK CHICS	-	5.00
SPACE BABES	-	6.70
SPEEDY TURTLES	-	7.90
STINGRAYS	-	7.10
SWEETHEARTS	-	6.90
TERMINATORS	-	8.30
TOUCANS	-	6.70
TWISTING TORNADOES	-	5.50
...		

Executabilul obținut în urma compilării va avea numele lanParty, iar regula de rulare va fi:

./lanParty c.in d.in r.out

Reguli de trimitere

- Puteți încărca mai multe soluții, se va lua în considerare ultima soluție încărcată, termen limită x
- Arhiva temei va avea numele GrupaSerie_Nume_Prenume_TemaNr.zip și va fi încărcată pe moodle.
- Arhiva trimisă conține (direct în rădăcină):
 1. Fișierul sau fișierele (.cpp și .hpp) cu codul programului;
 2. Makefile-ul (cu regulile make build și make clean). Executabilul generat trebuie să se numească lanParty;
 3. fișierul README în care va fi descrisă soluția problemei.

Restricții

- Implementarea se va face folosind limbajul C++;
- Implementarea se va face folosind liste. Nu e permisă implementarea cu vector a stivelor, cozii și a listei cu echipe participante. Se recomandă folosirea pointerilor și eliberarea memoriei;
- Se va evita pe cât posibil duplicarea nodurilor din lista pentru abateri grave de la această cerință se vor scădea până la 5 puncte.
- Nu se acceptă implementări cu tipuri de date cu memorie alocată static (se acceptă numai variabile locale de tip buffer pentru stocare temporară înainte de alocare);
- Se va depuncta lucrul nemodularizat (fără clase cu funcții membre). Operațiile pe structurile de date de tip arbore, coada, etc se fac folosind clase și funcții membre - Ex: adaugaNod, stergeNod, etc. Orice alte manipulări de date se fac, pe cât posibil, prin funcții specializate. ;
- Memoria trebuie eliberată. Dacă nu se respectă această cerință depunctarea este de până la 10/100 pct (restricție aplicabilă de la 60 de puncte în sus).
- Mențineți cel puțin un nivel minimal de aspect al codului și evitați inconsistența (indentare haotică, numeroase combinații de caractere de tip „leading/trailing whitespace“, numirea variabilelor și a funcțiilor în ordinea literelor din alfabet);
- Dacă soluția voastră nu compilează, dar ideea este bună și trimiteți o încercare de implementare, puteți primi până la 20/100 pct;
- Se va depuncta cu până la 15% din punctajul obținut implementarea care nu folosește tipuri de date specifice pentru entitățile din cerință (ex: Player)

