

AA - TEMA 2

- GIGEL ȘI MAFIOȚII -

Responsabili:

Teodor Milea, Matei Simtinică, Radu Nichita

Deadline soft: **24.01.2021**

Deadline hard: **24.01.2021**

CUPRINS

1	Plantarea Spionilor	3
1.1	Enunț	3
1.2	Cererea către Oracolul Antic	3
1.2.1	Date de intrare	3
1.2.2	Date de ieșire	3
1.2.3	Restricții și precizări	4
1.2.4	Exemplu	4
1.3	Descifrarea oracolului antic	4
1.3.1	Date de intrare	4
1.3.2	Date de ieșire	5
1.3.3	Exemplu	5
2	Investigațiile Familiilor Extinse	6
2.1	Enunț	6
2.2	Cererea către oracolul antic	6
2.2.1	Date de intrare	6
2.2.2	Date de ieșire	6
2.2.3	Restricții și precizări	6
2.2.4	Exemplu	7
2.3	Descifrarea oracolului antic	7
2.3.1	Date de intrare	7
2.3.2	Date de ieșire	7
2.3.3	Exemplu	7
3	Arestările Mafioților	9

3.1	Enunț	9
3.2	Cererea către oracolul antic	9
3.2.1	Date de intrare	9
3.2.2	Date de ieșire	9
3.2.3	Restricții și precizări	9
3.2.4	Exemplu	10
4	Bonus: Arestările mafioților tura doi	11
4.1	Enunț	11
4.2	Cererea către oracolul antic	11
4.2.1	Date de intrare	11
4.2.2	Date de ieșire	11
4.2.3	Restricții și precizări	11
4.2.4	Exemplu	12
4.3	Descifrarea oracolului antic	12
4.3.1	Date de intrare	12
4.3.2	Date de ieșire	12
4.3.3	Exemplu	12
5	Clarificări pentru folosirea Oracolului	13
6	Punctare	14
6.1	Checker	14
7	Format arhivă	15

1 PLANTAREA SPIONILOR

1.1 Enunț

Gigel a devenit un nou detectiv al Institutului Mistic de Criminalistică, care lucrează la desființarea rețelelor de crima organizată, cu ajutorul unui Oracol Antic. Acest Oracol poate răspunde oricărei întrebări, atât timp cât aceasta este enunțată ca o formulă în forma normal conjunctivă. Deși cunoașterea lui e infinită, este greu să comunici cu el dacă nu ai pregătirea potrivită. De aceea, Gigel speră să îl puteți voi ajuta. Va trebui să înțelegeți diferitele probleme pe care Gigel trebuie să le rezolve și să îl ajutați să comunice cu Oracolul.

Pentru ca munca voastră să aibă un impact de durată, va trebui să automatizați acest proces prin intermediul unor programe specifice pentru fiecare problemă în parte. Echipa de AA v-a pus deja la dispoziție un schelet de la care să plecați.

Prima problemă pe care Gigel trebuie să o rezolve în lupta cu crima organizată este plasarea unui număr de K spioni în N familii mafioate, astfel încât spionii să nu fie descoperiți. Pentru a evita compromiterea misiunii, va trebui să punem spionii astfel încât oricare 2 familii care se înțeleg să fie infiltrate de spioni diferiți.

1.2 Cererea către Oracolul Antic

Pentru a face o cerere la Oracolul Antic, trebuie să reduceți problema dată la problema SAT. Există mai multe modalități în care puteți să faceți acest lucru, iar implementarea checker-ului vă oferă libertatea de a vă alege voi ce reducere doriți să folosiți.

1.2.1 Date de intrare

Pe prima linie a fișierului `t1.in` se află 3 numere întregi N , M și K , reprezentând numărul familiilor de mafioți, numărul relațiilor dintre ele și spionii disponibili pentru misiune.

Pe următoarele M linii se află 2 numere întregi u și v , semnificând faptul că familiile u și v se înțeleg.

1.2.2 Date de ieșire

Prima linie a fișierului `t1.cnf` va conține stringul `"p cnf"` urmat de numerele V și F .

Următoarele F linii conțin variabilele pentru fiecare clauză din formula finală. Variabilele sunt despărțite de un spațiu, iar la final veți adăuga un 0, ce va reprezenta terminarea clauzei.

1.2.3 Restricții și precizări

- Variabilele nu pot conține 0, deci va trebui să începeți numirea variabilelor de la 1.
- Două familii care nu se înțeleg pot avea ori același spion, ori spioni diferiți.
- Un spion poate să infiltreze mai multe familii, atât timp cât se respectă condițiile anterioare.
- Se pot repartiza și mai puțin de K spioni, este esențial să nu fie mai mult de K spioni.

1.2.4 Exemplu

Exemplu 1	
t1.in	t1.cnf
4 4 1 1 2 2 3 2 4 3 4	p cnf <nr_variables> <nr_clauses> < var_1 > < var_2 >... < var_n > 0 ...

1.3 Descifrarea oracolului antic

Răspunsul pe care îl veți primi de la Oracol va conține doar alegerea variabilelor. Va trebui să faceți și o interpretare a acestui răspuns pe baza reducerii pe care ați ales-o.

1.3.1 Date de intrare

Pe prima linie a fișierului **r1.sol** se va afla răspunsul Oracolului:

- **True**, dacă formula voastră are soluție.
- **False**, dacă formula voastră nu are soluție.

În cazul în care formula are soluție, pe a doua linie va fi numărul **V**, reprezentat de numărul variabilelor formulei.

Pe ultima linie vor fi **V** numere reprezentând numele variabilelor din formula voastră. Aceste numere vor fi pozitive sau negative, cele pozitive reprezentând atribuirea valorii **True** acestui literal, iar cele negative reprezentând atribuirea valorii **False** literalului respectiv.

1.3.2 Date de ieșire

În cazul în care răspunsul oracolului este **False** fișierul **r1.out** va conține o singură linie cu textul *False*.

Altfel, pe prima linie va apărea textul *True*, iar pe a doua linie a fișierului **r1.out** vor fi **N** numere cu valori între 1 și **K**, reprezentând numărul spionului asignat fiecărei familii.

1.3.3 Exemplu

Exemplu 1		
r1.sol	r1.out	Explicație
True 5 -1 2 3 -4 5	True 1 2 3 2	Prima familie va fi repartizată spionului numărul 1. A doua familie va fi repartizată spionului numărul 2. A treia familie va fi repartizată spionului numărul 3. A patra familie va fi repartizată spionului numărul 2.

Observație: Interpretarea soluției oracolului depinde de rezolvarea voastră!

2 INVESTIGAȚIILE FAMILIILOR EXTINSE

2.1 Enunț

Gigel trebuie să descopere grupurile cele mai influente de familii din mafia locală, mai exact familiile extinse. Prin familie extinsă înțelegem un set de familii cu proprietatea că oricare două familii din set se înțeleg. Misiunea voastră este să îl ajutați pe Gigel să descopere aceste familii, urmând ca Institutul să însceneze flagranturi și să aresteze membrii acestor familii.

2.2 Cererea către oracolul antic

Această reducere este posibil să fie mai complicată de făcut, așa că vă încurajăm să porniți de la această reducere.

2.2.1 *Date de intrare*

Pe prima linie a fișierului **t2.in** se află 3 numere întregi **N**, **M** și **K**, reprezentând numărul familiilor de mafioți, numărul relațiilor dintre ele și respectiv dimensiunea familiei extinse căutate.

Pe următoarele **M** linii se află 2 numere întregi **u** și **v**, semnificând faptul că familiile **u** și **v** se înțeleg.

2.2.2 *Date de ieșire*

Pe prima linie a fișierului **t2.cnf** va conține stringul "*p cnf*" urmat de numerele **V** și **F**.

Următoarele **F** linii conțin variabilele pentru fiecare clauză din formula finală. Variabilele sunt despărțite de un spațiu, iar la final veți adăuga un 0, ce va reprezenta terminarea clauzei.

2.2.3 *Restricții și precizări*

- Variabilele nu pot conține 0, deci va trebui să începeți numirea variabilelor de la 1.
- HINT: Pentru această problemă, Gigel crede ca poate folosi acest **pergament**.

2.2.4 Exemplu

Exemplu 1	
t2.in	t2.cnf
4 4 1 1 2 2 3 2 4 3 4	p cnf <nr_variables> <nr_clauses> < var_1 > < var_2 >... < var_n > 0 ...

2.3 Descifrarea oracolului antic

2.3.1 Date de intrare

Pe prima linie a fișierului **r2.sol** se va afla răspunsul Oracolului:

- **True**, dacă formula voastră are soluție.
- **False**, dacă formula voastră nu are soluție.

În cazul în care formula are soluție, pe a doua linie va fi numărul **V**, reprezentat de numărul variabilelor formulei.

Pe ultima linie vor fi **V** numere reprezentând numele variabilelor din formula voastră. Aceste numere vor fi pozitive sau negative, cele pozitive reprezentând atribuirea valorii **True** acestui literal, iar cele negative reprezentând atribuirea valorii **False** literalului respectiv.

2.3.2 Date de ieșire

În cazul în care răspunsul oracolului este **False** fișierul **r2.out** va conține o singură linie cu textul *False*.

Altfel, pe prima linie va apărea textul *True*, iar pe a doua linie a fișierului **r2.out** vor fi **K** numere cu valori între 1 și **N**, reprezentând familiile care formează o familie extinsă de dimensiune **K**.

2.3.3 Exemplu

Exemplu		
r2.sol	r2.out	Explicație
True 5 -1 2 3 -4 5	True 2 3 5	Familiile 2, 3 și 5 formează o familie extinsă.

Observație: Interpretarea soluției oracolului depinde de rezolvarea voastră!

3 ARESTĂRILE MAFIOȚILOR

3.1 Enunț

În acest moment, cu spionii infiltrați și familiile extinse descoperite suntem gata să distrugem rețeaua infracțională, arestând familiile mafiotă. Din păcate, nu putem aresta toate familiile, deoarece Institutul nu are fonduri suficiente să organizeze atâtea flagranturi. Din fericire, nu este nevoie să arestăm toate familiile, este suficient să distrugem rețeaua mafiotă. Pentru a reuși, este nevoie ca familiile rămase în libertate să nu se înțeleagă între ele, nefiind astfel capabile să continue infracțiunile la scala actuală.

Misiunea lui Gigel este să descopere numărul **minim** de arestări necesare pentru a lăsa rețeaua doar cu familiile care nu se înțeleg.

3.2 Cererea către oracolul antic

Pentru a găsi rezultatul acestei problemei, va fi nevoie să formulați cereri multiple spre Oracol, deoarece el poate răspunde doar unor probleme de decizie, nu de optim. Scheletul pentru acest task este adaptat să vă permită să efectuați oricâte cereri aveți nevoie pentru a rezolva problema.

3.2.1 Date de intrare

Pe prima linie a fișierului **t3.in** se află 2 numere întregi **N**, **M**, reprezentând numărul familiilor de mafioți și respectiv numărul relațiilor dintre ele.

Pe următoarele **M** linii se află 2 numere întregi **u** și **v**, semnificând faptul că familiile **u** și **v** se înțeleg.

3.2.2 Date de ieșire

Fișierul **r3.out** va conține o singură linie cu numerele familiilor ce trebuie arestate.

3.2.3 Restricții și precizări

- Variabilele nu pot conține 0, deci va trebui să începeți numirea variabilelor de la 1.
- Sunt posibile mai multe soluții, nu contează care dintre acestea este afișată.
- Pentru această problemă există mereu o soluție, chiar dacă presupune arestarea tuturor familiilor. Problema constă în minimizarea acestor arestări. Prin urmare, în cel mai rău caz vor fi **N** arestări.
- HINT: Încercați să reduceți această problemă la una dintre problemele anterioare.

3.2.4 Exemplu

Exemplu 1		
r3.in	r3.out	Explicație
4 4 1 2 2 3 2 4 3 4	2 3	Familiile 2 si 3 formează grupul de mărime minimă ce trebuie arestat.

Observație: Interpretarea soluției oracolului depinde de rezolvarea voastră!

4 BONUS: ARESTĂRILE MAFIOȚILOR TURA DOI

4.1 Enunț

Deși am reușit să găsim numărul minim de arestări necesare, a trebuit să deranjăm Oracolul de prea multe ori. Vom încerca acum să rezolvăm problema anterioară printr-o singură cerere către Oracol.

4.2 Cererea către oracolul antic

Pentru această cerință vom folosit conceptul de soft clauses. Până acum am rezolvat doar probleme SAT cu hard clauses, în care toate clauzele trebuie să fie satisfăcute. Ce putem să facem în schimb este să oferim posibilitatea Oracolului să nu satisfacă toate clauzele. O clauză ce poate să nu fie satisfăcută se numește clauză soft. Fiecare clauză soft are asociată o pondere care este folosită de oracol pentru a minimiza suma ponderilor clauzelor nesatisfăcute. Pentru această problemă trebuie să folosiți clauze soft pentru a face o singură cerere spre solver. Pentru mai multe detalii legate de clauze soft puteți intra [aici](#) (secțiunea **Weighted Partial Max-SAT input format**).

4.2.1 Date de intrare

Pe prima linie a fișierului **bonus.in** se află 2 numere întregi **N**, **M**, reprezentând numărul familiilor de mafioți și respectiv numărul relațiilor dintre ele.

Pe următoarele **M** linii se află 2 numere întregi **u** și **v**, semnificând faptul că familiile **u** și **v** se înțeleg.

4.2.2 Date de ieșire

Pe prima linie a fișierului **bonus.wcnf** va conține stringul "*p wcnf*" urmat de numerele **V**, **F** și **S + 1**, unde **S** este suma tuturor ponderilor clauzelor opționale.

Următoarele **F** linii o să fie câte o reprezentare pentru fiecare clauză. O clauză va fi reprezentată de ponderea corespunzătoare ei și de numele variabilelor despărțite de un spațiu, cu 0 la finalul liniei. În cazul în care clauza este una obligatorie, ponderea corespunzătoare clauzei este **S + 1**.

4.2.3 Restricții și precizări

- Variabilele nu pot conține 0, deci va trebui să începeți numirea variabilelor de la 1.

4.2.4 Exemplu

Exemplu 1	
bonus.in	bonus.wcnf
4 4	p wcnf <nr_variables> <nr_clauses> <sum_weights + 1> <weight_1 > <var_1> <var_2> ... <var_n> 0 ...
1 2	
2 3	
2 4	
3 4	

4.3 Descifrarea oracolului antic

4.3.1 Date de intrare

Pe prima linie a fișierului **bonus.sol** va fi numărul **V**, reprezentat de numărul variabilelor formulei.

Pe a doua linie vor fi **V** numere reprezentând numele variabilelor din formula voastră. Aceste numere vor fi pozitive sau negative, cele pozitive reprezentând atribuirea valorii **True** acestui literal, iar cele negative reprezentând atribuirea valorii **False** literalului respectiv.

4.3.2 Date de ieșire

Fișierul **bonus.out** va conține o singură linie cu numerele familiilor ce trebuie arestate.

4.3.3 Exemplu

Exemplu 1		
bonus.sol	bonus.out	Explicație
5 1 -2 -3 4 5	2 3	Familiile 2 si 3 formează grupul de mărime minimă ce trebuie arestat.

Observație: Interpretarea soluției oracolului depinde de rezolvarea voastră!

5 CLARIFICĂRI PENTRU FOLOSIREA ORACOLULUI

Să presupunem că în urma reducerii pe care ați făcut-o pentru o problemă aveți o formulă de tipul:

$$(x \vee y \vee \neg z \vee \neg w) \wedge (\neg x \vee z) \wedge (\neg y \vee w) \wedge (x \vee y \vee w) \wedge (\neg y \wedge \neg z)$$

Pentru a codifica această formulă într-un fișier de tipul DIMACS care să poată fi folosit de Oracol (SAT solver), va trebui să oferim fiecărei variabile câte un număr pozitiv de la 1 până la 4 (numărul total de variabile). Să spunem că alegem $x \rightarrow 1$, $y \rightarrow 2$, $z \rightarrow 3$, $w \rightarrow 4$. Atunci fișierul .cnf corespunzător acestei formule va fi

```
p cnf 4 5
1 2 -3 -4 0
-1 3 0
-2 4 0
1 2 4 0
-2 -3 0
```

Prima linie conține antetul obligatoriu *p cnf* urmat de numărul de variabile și numărul de clauze. Următoarele linii vor conține codificările pentru fiecare clauză. Acestea sunt formate dintr-un șir de numere semnificând variabilele din clauză, terminându-se cu cifra 0. Fiecare din numerele pentru clauze pot fi negative sau pozitive, un număr negativ reprezentând că variabila corespunzătoare numărului este negată în clauză, iar unul pozitiv că variabila nu este negată.

Fișierul pe care oracolul o să îl creeze va avea forma

```
True
4
1 -2 3 4
```

Acest fișier arată că formula poate fi satisfăcută și după oferă o alegere a variabilelor din formulă. Pentru cazul nostru, se observa ca alegând y ca *False*, și restul ca *True*, formula este satisfăcută.

6 PUNCTARE

- Punctajul temei este de 100 puncte, distribuit astfel:
 - **Plantarea Spionilor:** 30p
 - **Investigațiile Familiilor Extinse:** 20p
 - **Arestările Mafioților:** 40p
 - Comentarii și README: 10p
- Punctajul pe README și comentarii este condiționat de obținerea a unui punctaj strict pozitiv pe cel puțin un test.
- Se poate obține un bonus de 20p rezolvând problema **Arestările mafioților tura doi**. Acordarea bonusului **NU** este condiționată de rezolvarea celorlalte probleme. În total se pot obține 120 de puncte (**NU** se trunchiază).
- Vor exista mai multe teste pentru fiecare problemă în parte. Punctele pe teste sunt independente, punctajul pe un anumit test nefiind condiționat de alte teste.
- În fișierul README va trebui să descrieți soluția pe care ați ales-o pentru fiecare problemă, să precizați complexitatea pentru fiecare și alte lucruri pe care le considerați utile de menționat.
- Pentru a primi punctajul complet pe o problemă este nevoie să implementați atât cererea spre oracol, cât și descifrarea răspunsului, deoarece checker-ul verifică doar validitatea răspunsului final, permițându-vă astfel să aveți libertate deplină în alegerea reducerilor pe care doriți să le folosiți.

6.1 Checker

- Atât checkerul cât și scheletul sunt disponibile pe [aici](#).
- Arhiva se va trimite **OBLIGATORIU** pe [vmchecker](#), unde tema se va testa folosind un set de teste private.
- Pentru testarea locală, aveți disponibil un set de teste publice (de aceeași dificultate) pe pagina temei.
Înainte de a rula prima dată checker-ul, va trebui să rulați script-ul **install.sh** din arhivă.
- Pentru o descriere a utilizării checker-ului puteți rula comanda: `python3 checker.py help`.
- Pentru a rula un test custom pentru task-ul X, completați fișierul **custom.in** din folder-ul `/in/taskX`. În acest caz, checker-ul se va apela astfel: `python3 checker.py`

<limbaj> <task> custom.

- **Punctajul pe teste** este cel de pe vmchecker și se acordă rulând tema doar cu testele private.
- Checkerul verifică doar existența unui README cu denumire corectă și conținut nenul. **Punctajul final pe README și comentarii** se acordă la corectarea manuală a temei.
- Corectorii își rezervă dreptul de a scădea puncte pentru orice problemă găsită în implementare, dacă vor considera acest lucru necesar.
- Pentru citirea în Java se recomandă folosirea **BufferedReader**.

7 FORMAT ARHIVĂ

- Temele vor fi testate automat pe vmchecker. Acesta suportă temele rezolvate în C/C++ sau Java.
- Arhiva cu rezolvarea temei trebuie să fie **.zip**, având un nume de forma **Grupa_NumePrenume_Temaz.zip** (ex: 399CX_PuiuGigel_Temaz.zip).
- **ATENȚIE!** Tema va fi compilată și testată **DOAR pe Linux**.
- **ATENȚIE!** Pentru cei ce folosesc C/C++ **NU** este permisă compilarea cu opțiuni de optimizare a codului (O1, O2, etc.).
- **ATENȚIE!** Orice nerespectare a restricțiilor duce la pierderea punctajului (după regulile de mai sus).