PA - TEMA 1
- GIGEL ȘI NOILE SALE AVENTURI -

Responsabili:
Cristian Pătrașcu, Andrei Preda, Ioan Pop
Ionela Nicuță, Claudiu Dorobanțu

Deadline soft: **20.04.2022 23:55**
Deadline hard: **27.04.2022 23:55**

CUPRINS

1 PROBLEMA 1: GIGEL ȘI TABELELE WALSH

1.1 Enunț

Gigel vrea să își implementeze un canal secret de comunicație. După ce a făcut niște cercetări, el a descoperit că o modalitate bună de criptare este folosirea Tabelelor Walsh. Acestea au forma unei matrice pătratice, de dimensiuni $\mathbf{N}\times\mathbf{N}$, unde $\mathbf{N}$ este o putere a lui 2. Notăm W_1 tabelul de dimensiune 1×1, W_2 tabelul de dimensiune 2×2, W_4 tabelul de dimensiune 4×4 și așa mai departe. Tabelele Walsh se generează după următoarea regulă:

$$W_1 = \begin{bmatrix} 0 \end{bmatrix} \quad W_{2n} = \begin{bmatrix} W_n & W_n \\ W_n & \overline{W_n} \end{bmatrix}$$

unde $\overline{W_n}$ va conține elementele din W_n negate (1 devine 0, iar 0 devine 1). Observăm că așa vor arăta W_2, W_4 si W_8, generate pe baza relației de mai sus.

$$W_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \quad W_4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \quad W_8 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Având acest tabel, Gigel se întreabă: Pentru $\mathbf{K}$ perechi de poziții $(\mathbf{x}, \mathbf{y})$: ce valoare se află în tabelul Walsh la linia $\mathbf{x}$, coloana $\mathbf{y}$? (de exemplu, pentru: $\mathbf{x} = 3$ și $\mathbf{y} = 2 \implies 0$)

1.2 Date de intrare

Fișierul de intrare **walsh.in** conține pe prima linie numărul natural $\mathbf{N}$, ce reprezintă dimensiunea tabelei Walsh.

Pe a doua linie se află numărul natural $\mathbf{K}$, iar pe următoarele $\mathbf{K}$ linii se află perechi de numere naturale $(\mathbf{x}, \mathbf{y})$ $(1 \leqslant \mathbf{x}, \mathbf{y} \leqslant \mathbf{N})$, câte o pereche pe o linie. Valorile scrise pe aceeași linie sunt seprate prin câte un spațiu.

1.3 Date de ieșire

Fișierul de ieșire **walsh.out** va conține $\mathbf{K}$ linii. Pe cea de-a $\mathbf{i}$-a linie va fi scrisă valoarea aflată pe linia $\mathbf{x}$ și coloana $\mathbf{y}$ pentru cea de-a $\mathbf{i}$-a pereche din fișierul de intrare.

1.4 Restricții și precizări

- $1 \leqslant \mathbf{N} \leqslant 2^{30}$, **N** putere a lui 2
- $1 \leqslant \mathbf{x}, \mathbf{y} \leqslant \mathbf{N}$
- $1 \leqslant \mathbf{K} \leqslant 10^5$

1.5 Testare și punctare

- Punctajul maxim este de **25** puncte.
- Timpul de execuție:
  - C/C++: **0.5 s**
  - Java: **2.5 s**
- Sursa care conține funcția **main** trebuie obligatoriu denumită: **walsh.c**, **walsh.cpp** sau **Walsh.java**.

1.6 Exemple

1.6.1 Exemplul 1

Exemplul 1		
walsh.in	walsh.out	Explicație
4 5 1 3 4 2 3 3 1 2 2 1	0 1 1 0 0	Pe linia 1, coloana 3 valoarea este 0 Pe linia 4, coloana 2 valoarea este 1 Pe linia 3, coloana 3 valoarea este 1 Pe linia 1, coloana 2 valoarea este 0 Pe linia 2, coloana 1 valoarea este 0

2 PROBLEMA 2: GIGEL ȘI CUVINTELE CRIPTATE

2.1 Enunț

După ce a înțeles tabelele Walsh cu ajutorul vostru, acum vrea să treacă la pasul următor al implementării canalului său de comunicație. În timpul căutarilor sale pe internet, a găsit o listă de **N** cuvinte criptate, ce conțin doar litere ale alfabetului englez și, cum este pasionat și de statistică, vrea să raspundă la următoarea întrebare:

Care este numărul maxim de cuvinte din listă astfel încât, în urma concatenării lor, să existe o litera dominantă? Dacă nu se poate alege niciun cuvant, se va afișa -1.

O literă este dominantă într-un șir de caractere dacă numărul de apariții ale acelei litere este strict mai mare decât jumatate din lungimea șirului.

2.2 Date de intrare

Pe prima linie a fișierului **statistics.in** se află **N**.

Pe următoarele **N** linii se află câte un cuvânt din lista lui Gigel.

2.3 Date de ieșire

În fișierul **statistics.out** se va afla pe prima linie numărul maxim de cuvinte din listă, având o literă dominantă.

2.4 Restricții și precizări

- $1 \leqslant \mathbf{N} \leqslant 10^5$
- $1 \leqslant \mathbf{L} \leqslant 10^5$, unde **L** este suma lungimilor cuvintelor din listă

2.5 Testare și punctare

- Punctajul maxim este de **25** puncte.
- Timpul de execuție:
  - C/C++: **0.5 s**
  - Java: **1.5 s**
- Sursa care conține funcția **main** trebuie obligatoriu denumită: **statistics.c**, **statistics.cpp** sau **Statistics.java**.

2.6 Exemple

2.6.1 Exemplul 1

Exemplul 1		
statistics.in	statistics.out	Explicație
4 abbc cbb aa ab	3	Alegem cuvintele abbc, cbb si ab. În total sunt 5 litere b și lungimea totala este 9. $5 > 9/2$

2.6.2 Exemplul 2

Exemplul 2		
statistics.in	statistics.out	Explicație
2 ab ab	-1	După concatenare, observăm ca nu există niciun caracter dominant.

3 PROBLEMA 3: GIGEL ȘI PRINEL

3.1 Enunț

Gigel trebuie sa aibă grijă de fratele său mai mic, Prinel. Pentru a îl ține ocupat, în timp ce el se uită la meciul naționalei de fotbal a României, îi dă o lista **a** de **N** numere, inițial toate fiind **1**, pe care poate aplica următoarea operație: alege un număr **a[i]** și un număr **x**, divizor al lui **a[i]**, apoi înlocuiește **a[i]** cu **a[i]** + **x**.

Scopul este să transforme numerele din **a** în numerele dintr-o altă listă **target** dată, tot de lungime **N**, până se termină meciul de fotbal, folosind operația anterioară de maxim **K** ori.

Gigel îi mai dă și o listă **p**, **p[i]** = numărul de puncte obținute dacă **a[i] = target[i]** la finalul operațiilor.

Cum România vrea să obțină cât mai multe puncte în calificări, și Prinel vrea să obțină cât mai multe puncte și se intreabă care este numărul maxim de puncte pe care le poate obține?

3.2 Date de intrare

Pe prima linie a fișierului **prinel.in** se află **N** și **K**, separate prin spațiu.
Pe a doua linie se află vectorul **target** ce conține **N** elemente, separate prin spațiu.
Pe a treia linie se află vectorul **p** ce conține **N** elemente, separate prin spațiu.

3.3 Date de ieșire

În fișierul **prinel.out** se va afla numărul maxim de puncte ce pot fi obținute.

3.4 Restricții și precizări

- **N** $\leqslant 10^3$
- **K** $\leqslant 10^6$
- **target[i]** $\leqslant 10^5$
- **p[i]** $\leqslant 10^6$

3.5 Testare și punctare

- Punctajul maxim este de **30** puncte.
- Timpul de execuție:
  - C/C++: **1 s**
  - Java: **1 s**
- Sursa care conține funcția **main** trebuie obligatoriu denumită: **prinel.c**, **prinel.cpp** sau **Prinel.java**.

3.6 Exemple

3.6.1 Exemplul 1

Exemplul 1		
prinel.in	prinel.out	Explicație
4 4 1 7 5 2 2 6 5 2	9	Pentru **target[0]** nicio operație. Pentru **target[1]** 4 operații : $1 \to 2 \to 4 \to 6 \to 7$ Pentru **target[2]** 3 operații: $1 \to 2 \to 4 \to 5$ Pentru **target[3]** o operație: $1 \to 2$ Alegem **target[0]**, **target[2]** și **target[3]**, obținând $2 + 5 + 2 = 9$

4 PROBLEMA 4: GIGEL SI CHEIA DE LA PORTOFEL

4.1 Enunț

După ani de minat criptomonede, Gigel s-a decis că în sfârșit e timpul să le folosească. Când a vrut să acceseze portofelul, a realizat că are nevoie de o cheie **K** formată din litere mici ale alfabetului englez, pe care și-o notase cu mulți ani în urmă pe o bucată de hârtie. După lungi căutări, a găsit hârtia salvatoare, însă o parte din caractere se șterseseră din cauza trecerii ireversibile a timpului.

Pe spatele foii mai era scris un șir **S**, despre care Gigel știe că are următoarele proprietăți:

- **S** este un subșir al lui **K**
- **K** conține doar litere din **S**

Gigel a scris cheia **K** pe o altă bucată de hârtie, punand caracterul **?** în fiecare din pozițiile unde lipsea un caracter.

Înainte de a se apuca de bruteforce pentru a găsi cheia punând litere în loc de toate caracterele **?**, Gigel vrea să știe de câte ori apare subșirul **S** în toate cheile **K** posibile. Deoarece rezultatul poate fi mare, se dorește afișarea lui modulo $10^9 + 7$.

Notă: Un subșir al unui șir este format din elemente (nu neapărat consecutive) ale șirului respectiv, în ordinea în care acestea apar în șir.

4.2 Date de intrare

Pe prima linie a fișierului **crypto.in** vor fi **N** și **L**, separate prin spațiu.
Pe a doua linie va fi cheia **K**, cu **?** în locul caracterelor lipsă.
Pe a treia linie va fi subșirul **S**.

4.3 Date de ieșire

În fișierul **crypto.out** se va afla un număr, reprezentând numărul de apariții ale subșirului **S** în cheile **K**.

4.4 Restricții și precizări

- $1 \leqslant \mathbf{N} \leqslant 10^5$, lungimea cheii **K**
- $1 \leqslant \mathbf{L} \leqslant 10$, lungimea subșirului **S**

4.5 Testare și punctare

- Punctajul maxim este de **35** puncte.
- Timpul de execuție:
  - C/C++: **1.5 s**
  - Java: **2.5 s**
- Sursa care conține funcția **main** trebuie obligatoriu denumită: **crypto.c**, **crypto.cpp** sau **Crypto.java**.

4.6 Exemple

4.6.1 Exemplul 1

Exemplul 1		
crypto.in	crypto.out	Explicație
4 2 ?x?y xy	11	Șirurile posibile sunt: xxxy → 3 subșiruri xy xxyy → 4 subșiruri xy yxxy → 2 subșiruri xy yxyy → 2 subșiruri xy $3+4+2+2=11$

5 PROBLEMA 5 (BONUS): GIGEL REGELE COMERȚULUI

5.1 Enunț

După toate încercările prin care a trecut de-a lungul temelor de la PA, Gigel a devenit cel mai înțelept om și a fost ales regele regatului Gigeland. În acest moment, regatul se confruntă cu o problemă ce necesită o rezolvare imediată.

Regatul are o rețea comercială ce constă în **N** orașe aflate pe o dreaptă. Orașul **i** se află la coordonata **coord[i]**. Un oraș poate face comerț doar cu orașele vecine cu el (cel mai din stânga și cel mai din drepta au un singur vecin). Bineînțeles, pentru ca o rută comercială sa fie activă, are nevoie de negustori, mai exact, un număr de negustori egal cu distanța dintre cele două orașe. Pentru ca un oraș să fie activ, este nevoie ca ambele sale rute să fie active. Din păcate, în regat nu există suficienți negustori pentru a se ocupa de toate rutele comerciale.

Cum, după cum am spus, Gigel este cea mai înțeleaptă persoană din regat, bineînțeles că el va rezolva problema. Cum în fiecare zi numărul de negustori din regat poate să crească sau să scadă, el vrea să răspundă la **Q** intrebari de genul:

Având **M** negustori disponibili, care este numărul maxim **X**, astfel încât oricum am alege **X** orașe, negustorii disponibili pot fi distribuiți pe rutele comerciale astfel încât cele **X** orașe să fie active comercial?

Observatie: Dacă o rută este activă, este activă pentru ambele orașe pe care le leagă.

5.2 Date de intrare

Pe prima linie a fișierului **regele.in** se află **N**, numărul de orașe.

Pe a doua linie se află **N** numere, reprezentând coordonatele orașelor, separate prin spațiu, **sortate crescător**.

Pe a treia linie se află **Q**, numărul de întrebări.

Pe urmatoarele **Q** linii se află un număr **M** de negustori.

5.3 Date de ieșire

În fișierul **regele.out** vor fi scrise **Q** linii a câte un număr, reprezentând numărul maxim **X**, cu proprietatea din enunț.

5.4 Restricții și precizări

- $\mathbf{Q} \leqslant 5 \cdot 10^5$.
- $\mathbf{N} \leqslant 2000$.
- $\mathbf{coord[i]} \leqslant 10^9$.

5.5 Testare și punctare

- Punctajul maxim este de **25** puncte.
- Timpul de execuție:
  - C/C++: **2 s**
  - Java: **3.5 s**
- Sursa care conține funcția **main** trebuie obligatoriu denumită: **regele.c**, **regele.cpp** sau **Regele.java**.

5.6 Exemple

5.6.1 Exemplul 1

Exemplul 1		
regele.in	regele.out	Explicație
6 2 5 10 12 17 19 3 15 10 20	2 1 6	**2** → daca **X** ar fi 3, pentru activarea orașelor de la pozițiile 5, 12 și 17 este nevoie de 17 negustori. Orice combinație de 2 orașe poate fi activată folosind 15 negustori. **1** → orice oraș poate fi activat daca avem 10 negustori. Activarea simultană a orașelor de la coordonatele 5 și 17 nu este posibilă cu 10 negustori (este nevoie de 15). **6** → pentru activarea tuturor orașelor este nevoie de 17 negustori.

6 PUNCTARE

- Punctajul temei este de **125** puncte, distribuit astfel:
  - Problema 1: **25p**
  - Problema 2: **25p**
  - Problema 3: **30p**
  - Problema 4: **35p**
  - 5 puncte vor fi acordate pentru comentarii și README.
  - 5 puncte vor fi acordate automat de checker pentru coding style. Totuși, la corectarea manuala se pot aplica **depunctari de pană la 20 de puncte** pentru **coding style neadecvat**.

Punctajul pe README, comentarii și coding style este condiționat de obținerea unui punctaj strict pozitiv pe cel puțin un test.

Se poate obține un **bonus** de **25p** rezolvând problema **Gigel regele comerțului**. Acordarea bonusului **NU** este condiționată de rezolvarea celorlalte probleme. În total se pot obține 150 de puncte (**NU** se trunchiază).

Pentru detalii puteți să vă uitați și peste regulile generale de trimitere a temelor.

- O temă care **NU** compilează va fi punctată cu 0.
- O temă care **NU** trece niciun test pe vmchecker va fi punctată cu 0.
- 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.
- Fiecare problemă va avea o limită de timp pe test (precizată mai jos și pe pagina cu enunțul). Dacă execuția programului pe un test al acelei probleme va dura mai mult decât limita de timp, veți primi automat o puncte pe testul respectiv și execuția va fi întreruptă.
- 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.

6.1 Checker

- Arhiva se va trimite 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 cu resurse a temei.
- Checkerul se poate rula fără niciun parametru, caz în care va verifica toate problemele. De asemenea, se mai poate rula cu un parametru pentru a rula o anumită problemă:

```
./check.sh <1 | 2| 3 | 4 | 5>
./check.sh <walsh | statistics | prinel | crypto | regele >
./check.sh cs
```

- **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.
- La corectarea manuală se poate depuncta pentru **erori de coding style** care nu sunt semnalate de checker.
- Corectorii își rezervă dreptul de a scădea puncte pentru orice problemă găsită în implementare, dacă vor considera acest lucru necesar.
- Pentru citirea in Java se recomanda folosirea **BufferedReader**.

7 FORMAT ARHIVĂ

• Temele pot fi testate automat pe vmchecker. Acesta suportă temele rezolvate în C/C++ și Java. Dacă doriți să realizați tema în alt limbaj, trebuie să-i trimiteți un email lui Traian Rebedea (traian.rebedea@cs.pub.ro), în care să îi cereți explicit acest lucru.
• Arhiva cu rezolvarea temei trebuie să fie **.zip**, având un nume de forma **Grupa_NumePrenume_Tema1.zip** (ex: 399CX_PuiuGigel_Tema1.zip) și va conține:

- Fișierul/fișierele sursă
- Fișierul **Makefile**
- Fișierul **README** (fără extensie)

• Fișierul pentru make trebuie denumit obligatoriu **Makefile** și trebuie să conțină următoarele reguli:

- **build**, care va compila sursele și va obține executabilele
- **run-p1**, care va rula executabilul pentru problema 1
- **run-p2**, care va rula executabilul pentru problema 2
- **run-p3**, care va rula executabilul pentru problema 3
- **run-p4**, care va rula executabilul pentru problema 4
- **run-p5**, care va rula executabilul pentru problema bonus (**doar dacă** ați implementat și bonusul)
- **clean**, care va șterge executabilele generate

• **ATENȚIE!** Funcția **main** din rezolvarea unei probleme se va găsi într-o sursă ce trebuie obligatoriu denumită astfel:

- **walsh.c**, **walsh.cpp** sau **Walsh.java** - pentru problema 1
- **statistics.c**, **statistics.cpp** sau **Statistics.java** - pentru problema 2
- **prinel.c**, **prinel.cpp** sau **Prinel.java** - pentru problema 3
- **crypto.c**, **crypto.cpp** sau **Crypto.java** - pentru problema 4
- **regele.c**, **regele.cpp** sau **Regele.java** - pentru problema 5

• **ATENȚIE!** Tema va fi compilată și testată **DOAR pe Linux**.
• **ATENȚIE!** Numele regulilor și a surselor trebuie să fie exact cele de mai sus. Absența sau denumirea diferită a acestora va avea drept consecință obținerea a 0 puncte pe testele asociate problemei rezolvate de regula respectivă.
• **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).

8 LINKS

- Regulament general PA
- Google C++ Style Guide
- Google Java Style Guide
- Debugging și Structuri de Date