

# PA - TEMA 1 - GREEDY & DP

Responsabili:

Cristian Banu, Teodor Popescu, Radu Visan

Deadline soft: **7 aprilie 2019, ora 23:59**

Deadline hard: **14 aprilie 2019, ora 23:59**

## CUPRINS

1	Problema 1	3
1.1	Enunt . . . . .	3
1.2	Date de intrare . . . . .	3
1.3	Date de iesire . . . . .	3
1.4	Restrictii si precizari . . . . .	3
1.5	Testare si punctare . . . . .	3
1.6	Exemple . . . . .	4
1.6.1	Exemplu 1 . . . . .	4
2	Problema 2	5
2.1	Enunt . . . . .	5
2.2	Date de intrare . . . . .	5
2.3	Date de iesire . . . . .	5
2.4	Restrictii si precizari . . . . .	5
2.5	Testare si punctare . . . . .	5
2.6	Exemple . . . . .	6
2.6.1	Exemplu 1 . . . . .	6
3	Problema 3	7
3.1	Enunt . . . . .	7
3.2	Date de intrare . . . . .	7
3.3	Date de iesire . . . . .	7
3.4	Restrictii si precizari . . . . .	7
3.5	Testare si punctare . . . . .	7
3.6	Exemple . . . . .	8
3.6.1	Exemplu 1 . . . . .	8
3.6.2	Exemplu 2 . . . . .	8

4	Bonus	9
4.1	Enunt . . . . .	9
4.2	Cerinta . . . . .	9
4.3	Date de intrare . . . . .	9
4.4	Date de iesire . . . . .	9
4.5	Restrictii si precizari . . . . .	9
4.6	Testare si punctare . . . . .	9
4.7	Exemple . . . . .	10
4.7.1	Exemplu 1 . . . . .	10
5	Punctare	11
5.1	Checker . . . . .	11
6	Format arhivă	12
7	Links	12

## PROBLEMA 1

### Enunt

Se da un sir de  $N$  numere pozitive. Tuzgu si Ritza au la dispozitie acest sir si vor sa joace urmatorul joc: jucatorul la mutare are dreptul sa aleaga oricare dintre numerele ramase in sir si sa il elimine. Definim scorul unui jucator ca fiind suma numerelor eliminate de respectivul jucator pe parcursul jocului. Scopul fiecarui jucator este sa maximizeze diferenta dintre scorul sau si scorul adversarului.

Stiind ca Tuzgu si Ritza muta alternativ si ambii jucatori joaca optim, se cere determinarea diferentei maxime pe care o poate obtine Tuzgu in final.

### Date de intrare

Pe prima linie a fisierului **p1.in** se afla un numar intreg  $N$ .

Pe urmatoarea linie se afla  $N$  numere intregi  $v_1, v_2, v_3, \dots, v_N$  reprezentand cele  $N$  numere pe care le folosesc cei doi pe parcursul jocului.

### Date de iesire

In fisierul **p1.out** se va scrie diferenta maxima dintre scorul lui Tuzgu si scorul Ritzei, daca cei doi jucatori joaca optim.

### Restrictii si precizari

- $1 \leq N \leq 10^5$
- $0 \leq v_1, v_2, v_3, \dots, v_N \leq 10^9$
- Tuzgu incepe jocul
- Jocul se termina atunci cand au fost eliminate toate cele  $N$  elemente din sir
- Pentru teste in valoare de 15 puncte,  $1 \leq N \leq 10^3$

### Testare si punctare

- Punctajul maxim este de 30 puncte.
- Timpul de executie:
  - C/C++: 0.3 s
  - Java: 0.6 s
- Sursa care contine functia **main** trebuie obligatoriu denumita: **p1.c**, **p1.cpp** sau **P1.java**.

## Exemple

*Exemplu 1*

Exemplu 1		
p1.in	p1.out	Explicatie
3 1 5 3	3	<p>Tuzgu are la dispozitie sirul [1, 5, 3] si alege numarul 5, pe care il elimina.</p> <p>Ritza are la dispozitie sirul [1, 3] si alege numarul 3.</p> <p>Tuzgu are la dispozitie sirul [1] si alege numarul 1, iar jocul se termina.</p> <p>Tuzgu a obtinut scor 6, iar Ritza scor 3, diferenta maxima posibila pentru Tuzgu fiind egala cu 3.</p>

## PROBLEMA 2

### Enunt

Tuzgu si Ritza vor sa joace un nou joc, care opereaza tot asupra unui sir de  $N$  numere intregi pozitive. Inainte de startul jocului, Tuzgu este obligat sa elimine exact  $K$  numere din sir. Dupa aceasta operatie, cei doi joaca exact jocul descris la problema anterioara.

Stiind ca Tuzgu doreste sa elimine in mod optim  $K$  numere pentru a maximiza diferenta dintre scoruri, care este diferenta maxima dintre scorul sau si scorul Ritzei pe care acesta o poate obtine, dupa eliminarea celor  $K$  numere?

### Date de intrare

Pe prima linie a fisierului **p2.in** se afla doua numere intregi  $N$ , respectiv  $K$ .

Pe urmatoarea linie se afla  $N$  numere intregi  $v_1, v_2, v_3, \dots, v_N$  reprezentand cele  $N$  numere pe care le folosesc cei doi pe parcursul jocului.

### Date de iesire

In fisierul **p2.out** se va scrie diferenta maxima dintre scorul lui Tuzgu si scorul Ritzei, daca cei doi jucatori joaca optim, in urma eliminarii a exact  $K$  numere de catre Tuzgu, inainte de startul jocului.

### Restrictii si precizari

- $1 \leq N, K \leq 10^3$
- $0 \leq v_1, v_2, v_3, \dots, v_N \leq 10^9$
- La fel ca si la prima cerinta, Tuzgu incepe jocul, iar acesta se termina atunci cand au fost eliminate toate cele  $N$  elemente din sir

### Testare si punctare

- Punctajul maxim este de 35 puncte.
- Timpul de executie:
  - C/C++: 0.3 s
  - Java: 0.6 s
- Sursa care contine functia **main** trebuie obligatoriu denumita: **p2.c**, **p2.cpp** sau **P2.java**.

## Exemple

*Exemplu 1*

Exemplu 1		
p2.in	p2.out	Explicatie
3 1 1 5 3	4	Raspunsul este 4, deoarece in strategia optima, Tuzgu elimina numarul 3 din sir, pastrand astfel doar [1,5]. Apoi jocul incepe, acesta extrage numarul 5 din sir, iar Ritza extrage numarul 1, urmand ca jocul sa se termine. Scorul lui Tuzgu este 5, scorul Ritzei este 1, iar diferenta maxima pe care o poate obtine Tuzgu este 4. Daca ar fi eliminat numarul 5, atunci ar fi obtinut diferenta 2, iar daca ar fi eliminat numarul 1, ar fi obtinut, din nou, diferenta 2.

## PROBLEMA 3

### Enunt

Tuzgu si Ritza vor sa joace un al 3-lea joc. Acest joc seamana foarte mult cu jocul de la prima cerinta, singura diferenta fiind ca jucatorul la mutare are voie sa elimine ori elementul cel mai din stanga al sirului, ori elementul cel mai din dreapta.

Care este diferenta maxima pe care o poate obtine Tuzgu in acest caz?

### Date de intrare

Pe prima linie a fisierului **p3.in** se afla un numar intreg **N**.

Pe urmatoarea linie se afla **N** numere intregi  $v_1, v_2, v_3, \dots, v_N$  reprezentand cele **N** numere pe care le folosesc cei doi pe parcursul jocului.

### Date de iesire

In fisierul **p3.out** se va scrie diferenta maxima dintre scorul lui Tuzgu si scorul Ritzei, daca cei doi jucatori joaca optim.

### Restrictii si precizari

- $1 \leq N \leq 10^3$
- $0 \leq v_1, v_2, v_3, \dots, v_N \leq 10^9$
- Tuzgu incepe jocul
- Jocul se termina atunci cand au fost eliminate toate cele **N** elemente din sir

### Testare si punctare

- Punctajul maxim este de **35** puncte.
- Timpul de executie:
  - C/C++: **0.3 s**
  - Java: **0.6 s**
- Sursa care contine functia **main** trebuie obligatoriu denumita: **p3.c**, **p3.cpp** sau **P3.java**.

## Exemple

*Exemplu 1*

Exemplu 1		
p3.in	p3.out	Explicatie
3 1 5 3	-1	<p>Tuzgu incepe jocul, avand la dispozitie sirul [1,5,3].</p> <p>La prima mutare, va elimina numarul 3.</p> <p>Ritza are la dispozitie sirul [1,5] si va elimina numarul 5.</p> <p>Tuzgu ramane doar cu o singura optiune, numarul 1, pe care il va si elimina.</p> <p>Scorul lui Tuzgu este 4, iar cel al Ritzei este 5, deci in cel mai bun caz, Tuzgu poate obtine o diferenta de -1.</p>

*Exemplu 2*

Exemplu 2		
p3.in	p3.out	Explicatie
4 1 5 3 1	2	<p>Ordinea in care se fac mutarile, stiind ca cei doi jucatori joaca optim, este urmatoarea:</p> <ol style="list-style-type: none"> <li>1. Tuzgu elimina numarul 1 din dreapta</li> <li>2. Ritza elimina numarul 3 din dreapta</li> <li>3. Tuzgu elimina numarul 5 din dreapta</li> <li>4. Ritza elimina numarul 1</li> </ol> <p>In acest caz, scorul lui Tuzgu este 6, scorul Ritzei este 4 iar diferenta este 2.</p> <p>Daca Tuzgu ar fi eliminat numarul 1 din stanga, Ritza ar fi eliminat numarul 5, apoi Tuzgu ar fi eliminat 3, iar Ritza ar fi incheiat jocul, eliminand ultimul numar egal cu 1.</p> <p>In acest caz, scorul lui Tuzgu ar fi fost 4, iar cel al Ritzei ar fi fost 6, diferenta obtinuta de Tuzgu fiind -2, deci nu ar fi fost strategia optima.</p>



## BONUS

### Enunt

Dupa ce au jucat cele 3 jocuri prezentate anterior, a venit timpul ca Tuzgu si Ritza sa joace un ultim joc. Cu toate astea, acesta este identic cu cel de la prima cerinta, cu mentiunea ca, de data aceasta, Tuzgu va propune un sir de  $N$  numere pe care cei doi sa joace jocul. Bineinteles, stiind ca este liber sa aleaga ce sir doreste, Tuzgu isi propune sa gaseasca un sir pentru care, daca cei doi ar juca optim jocul de la prima cerinta, ar obtine diferenta dintre scorul sau si scorul Ritzei egala cu  $K$ .

Deoarece exista o infinitate de siruri de lungime  $N$  pentru care diferenta maxima dintre scorurile celor doi este egala cu  $K$ , impunem doua restrictii: sirul trebuie sa contina  $N$  numere intregi **distincte**, iar acestea pot lua valori din intervalul  $[1, V]$ .

### Cerinta

Se cere sa se determine numarul de siruri de  $N$  numere intregi **distincte**, cu valori in  $[1, V]$ , pentru care diferenta maxima dintre scorul lui Tuzgu si scorul Ritzei este exact  $K$ . Deoarece acest numar poate fi foarte mare, se cere restul impartirii acestui numar la 1000000007 ( $10^9 + 7$ ).

### Date de intrare

Pe prima si unica linie a fisierului **p4.in** se vor gasi trei numere intregi  $N$ ,  $K$ , respectiv  $V$ .

### Date de iesire

In fisierul **p4.out** se va scrie numarul de siruri care respecta proprietatile de mai sus, modulo 1000000007 ( $10^9 + 7$ ).

### Restrictii si precizari

- Pentru 25 de puncte,  $1 \leq N, K, V \leq 100$
- Pentru inca 10 puncte,  $1 \leq N, K, V \leq 10^5$

### Testare si punctare

- Punctajul maxim este de 35 puncte.
- Timpul de executie:

- C/C++: **0.6 s**
- Java: **1.2 s**
- Sursa care conține funcția **main** trebuie obligatoriu denumită:  
**p4.c**, **p4.cpp** sau **p4.java**.

## Exemple

### *Exemplu 1*

Exemplu 1		
p4.in	p4.out	Explicatie
3 2 4	12	Cele 12 siruri sunt: 1. [3,2,1] 2. [4,1,3] 3. [1,4,3] 4. [1,2,3] 5. [2,3,1] 6. [3,1,4] 7. [1,3,2] 8. [3,1,2] 9. [4,3,1] 10. [1,3,4] 11. [2,1,3] 12. [3,4,1]

## PUNCTARE

- Punctajul temei este de 160 de puncte, distribuit astfel:
  - Problema 1: 30p
  - Problema 2: 35p
  - Problema 3: 35p
  - Bonus: 35p
  - 12.5 puncte vor fi acordate pentru comentarii si README
  - 12.5 puncte vor fi acordate pentru coding style

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

Se poate obține un bonus de 35p rezolvând problema bonus. Acordarea bonusului **NU** este condiționată de rezolvarea celorlalte probleme. În total se pot obține 160 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 0 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.

## Checker

- Arhiva se va trimite pe **vmchecker**, unde tema se va testa folosind un set de teste private.
- Punctajul pe teste este cel de pe vmchecker și se acordă rulând tema pe același set de teste pus la dispoziție pe Moodle.
- 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 corectare 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.

## 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** 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
  - **clean**, care va șterge executabilele generate
  - **run-p4**, care va rula executabilul pentru problema bonus (doar dacă ați implementat și bonusul)
- **ATENȚIE!** Funcția **main** din rezolvarea unei probleme se va găsi într-o sursă ce trebuie obligatoriu denumită astfel:
  - **p1.c, p1.cpp** sau **P1.java** - pentru problema 1
  - **p2.c, p2.cpp** sau **P2.java** - pentru problema 2
  - **p3.c, p3.cpp** sau **P3.java** - pentru problema 3
  - **p4.c, p4.cpp** sau **P4.java** - pentru problema 4
- **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).

## LINKS

- [Regulament general teme PA](#)

- [Google C++ Style Guide](#)
- [Google Java Style Guide](#)
- [Debugging și Structuri de Date](#)