

Problema Calculator

Fișier de intrare calculator.in
Fișier de ieșire calculator.out

Se dau N procese care trebuie rulate pe un calculator. Există M variabile care pot fi accesate de către aceste procese, numerotate de la 1 la M . Fiecare proces i (unde $1 \leq i \leq N$) accesează un interval continuu de variabile $[l_i, r_i]$, iar timpul necesar de execuție al procesului i este t_i .

Cât timp un proces rulează, acesta își salvează pe stiva de execuție toate variabilele necesare, iar la finalizarea lui le elimină de pe stivă. Două procese pot rula în paralel dacă și numai dacă nu au nicio variabilă în comun. Sistemul de operare va planifica procesele astfel încât timpul total necesar executării tuturor să fie minim. O astfel de planificare, care atinge timpul minim posibil, poartă numele de *scenariu optim de execuție*.

O variabilă este considerată *critică* dacă există cel puțin un scenariu optim de execuție în care aceasta petrece cel mai mult timp pe stivă dintre toate variabilele (într-un scenariu pot exista mai multe variabile critice simultan).

Cerințe

Dat fiind un număr $C \in \{1, 2, 3\}$, reprezentând tipul cerinței:

Cerința 1. Să se determine timpul minim necesar pentru a rula toate cele N procese date.

Cerința 2. Pentru fiecare proces i (de la 1 la N), presupunem că îl eliminăm din sistem. Fie T_i timpul minim necesar pentru a rula restul de $N - 1$ procese. Să se calculeze și să se afișeze suma tuturor valorilor T_i , pentru i de la 1 la N .

Cerința 3. Pentru fiecare proces i (de la 1 la N), presupunem că îl eliminăm din sistem. Fie V_i numărul de variabile critice în contextul celorlalte $N - 1$ procese, conform definiției anterioare. Să se calculeze și să se afișeze suma tuturor valorilor V_i , pentru i de la 1 la N .

Generarea datelor

Pentru a evita citirea unui volum foarte mare de date, doar primele K procese vor fi citite din fișierul de intrare sub forma unor triplete (l_i, len_i, t_i) , pentru i de la 1 la K . Pentru cerința $C = 1$, se garantează că $N = K$, deci nu este necesară nicio generare suplimentară.

Pentru cerințele $C \in \{2, 3\}$, se citesc doar primele K procese. Restul proceselor, de la $K + 1$ până la N , se vor genera în programul vostru folosind următorul algoritm:

- 1: **pentru** $i \leftarrow K + 1, N$ **execută**
- 2: $l_i \leftarrow ((l_{i-1} \text{ xor } (a \cdot l_{i-K})) \bmod M) + 1$
- 3: $\text{len}_i \leftarrow ((\text{len}_{i-1} \text{ xor } (b \cdot \text{len}_{i-K})) \bmod M) + 1$
- 4: $t_i \leftarrow ((t_{i-1} \text{ xor } (c \cdot t_{i-K})) \bmod T) + 1$
- 5: **sfârșit pentru**

Notă: Operația logică pe biți XOR/„SAU exclusiv” este reprezentată în C/C++ prin operatorul ^.

Pentru toate procesele (indiferent dacă au fost citite sau generate), capătul drept al intervalului accesat de procesul i se va calcula după formula:

$$r_i = l_i + (\text{len}_i \bmod (M - l_i + 1))$$

Date de intrare

Fișierul de intrare calculator.in conține pe prima linie numerele C, N, M, K, T , urmate de constantele a, b și c , separate prin câte un singur spațiu. Pe următoarele K linii se vor afla câte trei numere: l_i, len_i și t_i , care reprezintă informațiile pentru primele K procese (unde $1 \leq i \leq K$).

Date de ieșire

Fișierul de ieșire calculator.out va conține pe o singură linie un singur număr natural, reprezentând răspunsul calculat conform cerinței C .

Restricții și precizări

- $1 \leq N, M \leq 5\,000\,000$

- $1 \leq K \leq \min(N, 100\,000)$
- $1 \leq T \leq 10\,000$
- $1 \leq l_i, \text{len}_i \leq M$ și $1 \leq t_i \leq T$, pentru orice $1 \leq i \leq K$
- $1 \leq a, b, c \leq 1\,000\,000\,000$
- **Atenție:** Calculele intermediare din algoritmul de generare a datelor pot depăși capacitatea de reprezentare a tipurilor întregi pe 32 de biți (de exemplu `int` sau `unsigned int`). Se recomandă și utilizarea tipurilor de date pe 64 de biți (de exemplu `long long`) unde este cazul.

#	Punctaj	Restricții
1	15	$C = 1, 1 \leq N, M \leq 2\,000, K = N$
2	15	$C = 1, 1 \leq N, M \leq 100\,000, K = N$
3	5	$C = 2, 1 \leq N, M \leq 200, K = N$
4	5	$C = 2, 1 \leq N, M \leq 2\,000, K = N$
5	10	$C = 2, 1 \leq N, M \leq 100\,000, K = N$
6	5	$C = 2, 1 \leq N, M \leq 2\,000\,000$
7	10	$C = 2$
8	5	$C = 3, 1 \leq N, M \leq 200, K = N$
9	5	$C = 3, 1 \leq N, M \leq 2\,000, K = N$
10	10	$C = 3, 1 \leq N, M \leq 100\,000, K = N$
11	5	$C = 3, 1 \leq N, M \leq 2\,000\,000$
12	10	$C = 3$

Exemple

calculator.in	calculator.out
1 3 5 3 10 1 1 1 1 3 5 3 3 4 4 2 2	9
2 3 5 3 10 1 1 1 1 3 5 3 3 4 4 2 2	20
3 3 5 3 10 1 1 1 1 3 5 3 3 4 4 2 2	3

Explicații

Pentru exemplele de mai sus, avem $N = 3$ procese și $M = 5$ variabile. Procesele sunt:

- Procesul 1: $l_1 = 1, r_1 = 1 + (3 \bmod 5) = 4$, timp $t_1 = 5$. Intervalul este $[1, 4]$.
- Procesul 2: $l_2 = 3, r_2 = 3 + (3 \bmod 3) = 3$, timp $t_2 = 4$. Intervalul este $[3, 3]$.
- Procesul 3: $l_3 = 4, r_3 = 4 + (2 \bmod 2) = 4$, timp $t_3 = 2$. Intervalul este $[4, 4]$.

Analizând interacțiunea dintre procese se observă că:

- Procesul 1 are variabile comune atât cu Procesul 2 (variabila 3), cât și cu Procesul 3 (variabila 4). Prin urmare, P_1 nu poate rula în paralel cu P_2 sau P_3 .
- Procesul 2 și Procesul 3 nu au nicio variabilă în comun, deci pot rula simultan.

Pentru primul exemplu ($C = 1$): Într-un scenariu optim, P_1 rulează singur (5 unități de timp), iar P_2 și P_3 rulează în paralel. Timpul necesar pentru grupul $\{P_2, P_3\}$ este $\max(t_2, t_3) = 4$. Timpul total minim este $5 + 4 = 9$.

Pentru al doilea exemplu ($C = 2$):

- Eliminăm P_1 : Rămân P_2 și P_3 , care rulează în paralel. $T_1 = \max(4, 2) = 4$.
- Eliminăm P_2 : Rămân P_1 și P_3 . Deoarece au variabila 4 în comun, trebuie să ruleze secvențial. $T_2 = 5 + 2 = 7$.
- Eliminăm P_3 : Rămân P_1 și P_2 . Au variabila 3 în comun, deci rulează secvențial. $T_3 = 5 + 4 = 9$.

Suma rezultatelor este $4 + 7 + 9 = 20$.

Pentru al treilea exemplu ($C = 3$):

- Fără P_1 : Timpul optim este 4. Variabila 3 stă pe stivă timp de 4 unități (cât rulează P_2), iar variabila 4 stă pe stivă 2 unități (cât rulează P_3). Singura variabilă critică este variabila 3. $V_1 = 1$.
- Fără P_2 : Timpul optim este 7. Procesele P_1 și P_3 rulează pe rând. Variabila 4 stă pe stivă în total $5 + 2 = 7$ unități, fiind singura care atinge acest timp maxim. $V_2 = 1$.
- Fără P_3 : Timpul optim este 9. Procesele P_1 și P_2 rulează pe rând. Variabila 3 stă pe stivă în total $5 + 4 = 9$ unități, fiind singura variabilă critică. $V_3 = 1$.

Suma valorilor V_i este $1 + 1 + 1 = 3$.