Problema Bt

Fişier de intrare	`bt.in`
Fişier de ieşire	`bt.out`

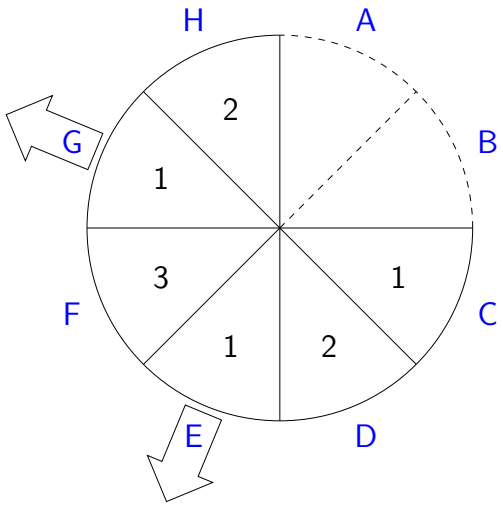
John și-a investit toate economiile în criptomonede. Acum încearcă să mai recupereze din bani, trecând pe dieta campionilor (brânză topită).

El are o cutie circulară cu N bucăți de brânză (de mai multe tipuri), iar în fiecare zi ia câte una din cutie. Acesta are grijă ca **înainte** și după ce ia o bucată din cutie, oricare două bucăți adiacente rămase să fie de alt tip.

John s-a apucat să numere în câte moduri poate să termine cutia.

Formal, se dă un vector circular $v_1, v_2, \ldots, v_N$. Vrem să scoatem din el câte un element până rămâne gol vectorul. În urma fiecărei scoateri, **inclusiv înaintea primei scoateri**, nu trebuie să existe 2 poziții consecutive în vector cu aceeași valoare în ele.

În exemplul din dreapta, bucățile de brânză de pe pozițiile A și B au fost deja luate. Singurele bucăți pe care le-am putea scoate acum ar fi E sau G. Dacă am scoate bucata C, atunci bucățile D si H, ambele de tipul 2, ar deveni vecine. Dacă l-am scoate pe D, C și E ar deveni vecine, deși sunt ambele de tipul 1, etc.



Cerință

Să se numere în câte moduri putem goli vectorul respectând regulile din enunț (modulo 1 000 000 007).

Un mod de a goli vectorul este definit de ordinea indicilor care părăsesc vectorul. De exemplu, pentru $N = 4$, sunt $4! = 24$ de moduri diferite de a goli vectorul (dintre care nu toate respecta regulile din enunț).

Date intrare

Pe prima linie a fișierului de intrare se găsește un singur număr natural N reprezentând lungime vectorului.

Pe a doua linie se găsesc N numere naturale, $v_1, v_2, \ldots, v_N$, reprezentând valorile din vector.

Date ieșire

În fişierul de ieşire se va afişa un singur număr, reprezentând numărul de moduri de a goli vectorul.

Problema are 2 moduri de punctare: vectorul este circular (pentru 100% din punctajul de pe un test), sau vectorul nu este circular - v_1 și v_N nu sunt considerați vecini (pentru 80% din punctaj).

Inițial, numărul afișat este comparat cu răspunsul corespunzător vectorului circular pentru 100% din punctajul testului.

Dacă răspunsurile diferă, numărul este apoi comparat cu răspunsul corespunzător vectorului normal (v_1 și v_N nu sunt considerați vecini) pentru 80% din punctajul testului.

Restricții

- $1 \leq N \leq 500$.
- $1 \leq v_i \leq N$.

#	Punctaj	Restricţii
1	10	$1 \leq N \leq 10$
2	10	$1 \leq N \leq 20$
3	30	$1 \leq N \leq 50$
4	50	Restricţii iniţiale.

Exemple

bt.in	bt.out	Explicații

4 1 2 1 2	0	În primul exemplu, orice am scoate am avea ulterior doi de 1 vecini, sau doi de 2 vecini, deci nu putem goli vectorul. Dacă am avea un delimitator între ultimul element și primul, răspunsul ar fi 8. Secvențele de indici corecte în acest caz ar fi: • $1, 2, 3, 4$ • $1, 2, 4, 3$ • $1, 4, 2, 3$ (vectorul ar arăta: $(1,2,1,2) \rightarrow (2,1,2) \rightarrow (2,1) \rightarrow (1) \rightarrow$ gol) • $1, 4, 3, 2$ • $4, 1, 2, 3$ • $4, 1, 3, 2$ • $4, 3, 1, 2$ • $4, 3, 2, 1$
8 1 2 1 3 1 2 1 3	1728	În al doilea exemplu, răspunsul corect dacă ar exista un delimitator este 6912.
4 1 2 3 4	24	În al treilea exemplu, deoarece orice element din vector este distinct, acestea pot fi scoase în orice ordine. Răspunsul pentru celălalt caz este tot $4! = 24$.
6 1 2 3 1 3 2	96	În al patrulea exemplu, răspunsul corect dacă ar exista un delimitator este 312.
1 1	1	În al cincilea exemplu, avem un singur element în vector, deci există o singură cale de a-l scoate. Răspunsul pentru celălalt caz este tot 1.

Problema Cactus

Fişier de intrare	`cactus.in`
Fişier de ieşire	`cactus.out`

~~Matei Nakayama~~ Mihai a primit un cactus.

Un cactus este un graf conex neorientat în care fiecare nod face parte din **cel mult un ciclu**. Se consideră un cactus cu N noduri și M muchii cu diverse lungimi. Acesta nu conține bucle (adică muchii între un nod și el insuși) sau muchii paralele (adică două sau mai multe muchii între o singură pereche de noduri).

Mihai vrea să elimine muchii din graful cactus, astfel încât să obțină un arbore cât mai ușor. Distanța dintre două noduri din arbore este egală cu suma lungimilor muchiilor de pe cel mai scurt drum dintre cele două noduri. Greutatea unui arbore este definită ca suma distanțelor dintre oricare două perechi neordonate de noduri distincte - perechea $(1, 2)$ este echivalentă cu perechea $(2, 1)$.

Ajutați-l pe Mihai să găsească **greutatea minimă** a unui arbore obținut prin eliminarea unor muchii din cactus.

Date intrare

Pe prima linie a fișierului de intrare se află două număre N, reprezentând numărul de noduri din graf, și M, reprezentând numărul de muchii din graf.

Pe fiecare dintre următoarele M linii se află trei numere x, y și z, reprezentând o muchie cu lungimea z între nodurile x și y.

Date ieșire

În fişierul de ieşire se va afişa un singur număr, reprezentând greutatea minimă a unui arbore ce se poate obține din cactusul inițial prin eliminarea unor muchii.

Restricții

- $1 \leq N \leq 100\,000$.
- $N - 1 \leq M \leq 200\,000$.
- $0 \leq z \leq 1\,000\,000\,000$
- **Se garantează că răspunsul este cel mult** 10^{18}.

#	Punctaj	Restricţii
1	4	Graful este un lanț (nu conține niciun ciclu și fiecare nod are grad cel mult 2).
2	6	Graful este un arbore (nu conține niciun ciclu).
3	12	$1 \leq N \leq 15$
4	25	$1 \leq N \leq 1\,000$
5	38	Graful este un ciclu (fiecare nod are grad 2).
6	15	Restricţii iniţiale.

Exemple

cactus.in	cactus.out	Explicații
6 6 1 2 8 1 3 2 3 2 1 1 4 3 4 5 2 2 6 4	80	Se elimină muchia (1 2).

12 14 1 2 7 2 3 3 1 3 7 3 4 2 4 5 5 5 6 10 6 4 3 4 7 4 7 8 2 8 9 5 9 10 8 10 11 1 11 7 9 10 12 3	787	Se elimină muchiile (1 2), (5 6), (9 10).

Problema Sirbun

Fişier de intrare	`sirbun.in`
Fişier de ieşire	`sirbun.out`

Un străbun get, Ziraxes, le-a dat dacilor liberi să rezolve o problemă de programare, aceasta fiind o activitate mai plăcută decât să care bolovani, pietricele şi nisip. Legenda spune că asupra elementelor unui şir A de numere naturale nenule se poate efectua următoarea operaţie:

Se alege un element A_i din şir şi un număr natural x şi se scade x din A_i, deci A_i devine $A_i - x$.

Şirul A se numeşte *bun* dacă aplicând operaţia de oricâte ori, elementele şirului A devin numere naturale **nenule** distincte. De exemplu, şirul $2, 3, 3, 5$ este bun deoarece scăzând 2 din al doilea element el devine $2, 1, 3, 5$ şi are elementele distincte, iar şirul $2, 2, 7, 2, 4$ nu este bun.

Cerinţă

Fiind dat un şir A format cu N elemente numere naturale nenule, determinaţi numărul subsecvenţelor din şir care sunt şiruri bune. O subsecvenţă a șirului este formată din elemente din şir aflate pe poziţii consecutive.

Date intrare

Pe prima linie a fişierului de intrare se află numărul N, iar pe a doua linie elementele şirului A.

Date ieșire

În fişierul de ieşire se va afişa numărul subsecvenţelor din şirul A care sunt şiruri bune.

Restricţii

- $1 \leq N \leq 100\,000$.
- $1 \leq A_i \leq N$.

#	Punctaj	Restricţii
1	19	$1 \leq N \leq 300$
2	20	$1 \leq N \leq 1500$
3	22	$1 \leq N \leq 7\,000$
4	17	$1 \leq N \leq 50\,000$
5	22	Restricţii iniţiale

Exemple

sirbun.in	sirbun.out
5 4 2 2 3 2	13

Explicație

Subsecvenţele bune sunt:

1. {4}
2. {2}
3. {2}
4. {3}
5. {2}
6. {4,2}
7. {4,2,2}
8. {4,2,2,3}
9. {2,2}
10. {2,2,3}
11. {2,3}
12. {2,3,2}
13. {3,2}