



Úlohy 2. kola letnej časti

Termín odoslania riešení tohto kola je pondelok 5.8.2019. Doprogramovávanie končí v pondelok 19.8.2019.

1. Reaktor v problémoch

12 b za popis, 8 b za program

Píše sa 31. december 1999. Frajerský Kóderi Systémov nezískali tender na napísanie programu pre tajnú jadrovú elektrárňu na matfýze. Jednalo sa o jeden z najdôležitejších programov – program na ovládanie žiarovky vnútri kontrolnej miestnosti. Len si predstavte tú galibu, keby bola v kontrolnej miestnosti tma. Alebo nebodaj, keby sa žiarovka zapla dvakrát. Brr – určite aj vás striaslo pri tejto myšlienke. Zákazku namiesto vás získal Klub Machrov Skoro-programátorov.

Mišofovi, vtedy ešte mladému a udatnému junákovi, to akosi nedalo. Vkradol sa preto pod rúškom noci do KMS servera a prezrel program, ktorý napísali. Chcel by program otestovať. Najprv však musí po sebe upratať, aby nikto nezistil, že tam lozil. Pomôžte mu zatiaľ napísať program, ktorým by otestoval to čo vytvorili KMS-áci!

Úloha

Na vstupe dostanete rozvrh časov, kedy má byť žiarovka zapnutá. Zistíte, či v tomto rozvrhu nastane situácia, kedy by žiarovka mala byť zapnutá viackrát naraz.

Formát vstupu

Na prvom riadku vstupu dostanete dve čísla - n a m . Číslo n ($1 \leq n \leq 10^6$) udáva dĺžku tmy v sekundách a číslo m ($1 \leq m \leq 500\,000$) je počet intervalov, kedy má byť žiarovka zapnutá. Každý z nasledujúcich m riadkov obsahuje dve čísla a a b ($0 \leq a < b \leq n$). Žiarovka by mala byť zapnutá v poloopenom intervale $[a, b)$, teda od času a do času $b - 1$ vrátane.

Formát výstupu

Vypíšte reťazec ANO ak je žiarovka zapnutá v nejakom momente dvakrát. Inak vypíšte reťazec NIE.

Príklad

vstup

```
5 3
0 1
1 2
2 3
```

výstup

```
NIE
```

vstup

```
10 3
3 5
0 2
1 2
```

výstup

```
ANO
```

2. Ako sa časy menia

12 b za popis, 8 b za program

Píše sa 31. jún 2019. Prešlo už viac ako 20 rokov odkedy ste nezískali tender na napísanie programu pre jadrovú elektrárňu na matfýze. Ale časy sa menia a tentoraz je šťastie na vašej strane. Ministerstvo Krajne Seri-ózných Projektov sa rozhodlo vykonať rutinnú 20 ročnú kontrolu bezpečnostných systémov kritických zariadení republiky a vy ste vyhrali tento tender! Hlava sa vám točí nad predstavou koľko korniet si budete vedieť kúpiť. Ale najprv práca, potom kornetá, hor sa do roboty!

Bezpečnostný systém funguje na báze Masívneho hesla, teda číselnej kombinácie, ktoré má uložené v pamäti a ktoré porovnáva s pokusmi od užívateľa. Vám však netrvalo dlho zistiť, že to kódil naozaj iba nejaký skoro-programátor. Pokus sa vyhodnotí ako nesprávny nie keď sa zadané heslo nezhoduje s Masívnym heslom, ale keď

obsahuje nejaké znaky navyše! To znamená, že na poradí znakov v hesle vôbec nezáleží a že napríklad prázdne heslo bude vždy správne.

Ministerstvo nebolo veľmi nadšené vaším zistením, ale odľahlo im, lebo vraj potenciálny útočník túto chybu nevedel použiť na zistenie Masívneho hesla. To totižto používajú úplne všade a ak by ho zistiť vedel, bola by šanca, že ho niekedy v minulosti získal alebo možno v blízkej budúcnosti získa. Toto by nemohli riskovať, museli by Masívne heslo všade zmeniť a komu sa to chce.

Dokážte, že sa mýlia a že sa táto chyba dá použiť na zistenie Masívneho hesla. Navrhnite program, ktorý toto Masívne heslo nájde.

Úloha

Táto úloha je interaktívna. Namiesto kompletného vstupu budete dostávať odpovede na vaše otázky. Heslo je podmnožina, teda niekoľko **rôznych** celých čísel od 1 po N , pričom Masívne heslo má dĺžku najviac M čísel.

Vaša otázka je tip na Masívne heslo. Odpoveď na vašu otázku bude kladná, pokiaľ je každý jeden prvok tejto otázky súčasťou Masívneho hesla. Teda otázka bude vyhodnotená záporne, pokiaľ obsahuje aspoň jeden prvok, ktorý nie je súčasťou Masívneho hesla. Vašou úlohou je zistiť Masívne heslo.

Môžete sa opýtať nanajvýš 111 111 otázok.

Formát vstupu

Na prvom riadku dostanete čísla N – najväčšie možné číslo v hesle a M – maximálnu dĺžku Masívneho hesla, pri čom vždy platí $(0 \leq M \leq N \leq 10^5)$.

Následne dostanete odpoveď na každú vašu otázku jedným riadkom obsahujúcim JOP v prípade kladného vyhodnotenia a NOP v prípade záporného.

Formát výstupu

Každá otázka, ktorú položíte, by mala byť vypísaná na jednom novom riadku. Riadok sa začína číslom K – dĺžkou hesla. Za ním nasleduje medzera a K medzerou oddelených čísel – vaša otázka. V prípade, že si myslíte, že poznáte Masívne heslo, vypíšete -1 a na nový riadok vypíšete toto heslo v rovnakom formáte, ako otázky (teda číslo K a za ním K medzerou oddelených čísel).

Pokiaľ sa opýtate viac ako 111 111 otázok, váš program bude nemilosrdne ukončený a v testovači sa to prejaví verdiktom „WA – Zlá odpoveď“ alebo dokonca „EXC – Chyba počas behu programu“.

Flushovanie

Programovacie jazyky štandardne používajú výstupný buffer, a až po jeho zaplnení program vypíše jeho obsah. Flushovanie vlastne znamená vypísanie (vyprázdnenie) tohto buffera. Vždy, keď vypíšete svoju otázku, musíte ešte flushnúť výstupný buffer. Inak sa váš ťah v skutočnosti nevypíše a nedostane sa k nášmu programu. V konečnom dôsledku potom dostanete hlášku „TLE – Prekročený časový limit“.

Ako flushovať?

Ak programujete v C/C++ a používate `printf()`, na to aby ste flushli jeho buffer, použijete `fflush(stdout)`. Ak používate `cout`, buffer sa flushuje automaticky po vypísaní konca riadku pomocou `cout<<endl`. Ak programujete v Pythone, používajte `print(vystup, flush=True)`. Ak programujete v Pascale použijete `flush(output)`.

Hodnotenie

Sú 4 testovacie sady, každá za 2 body. Platia v nich nasledovné obmedzenia:

Sada	1	2	3	4
Maximálne N	15	200	3 000	100 001

Navyše v druhej sade platí, že počet platných tajných hesiel je malý, teda hesiel ktoré neporušujú obmedzenie na maximálnu veľkosť Masívneho hesla.

V hodnotení popisu sa bude samozrejme klásť veľký dôraz na časovú zložitosť, ale tento krát ešte väčší na počet položených otázok. Odhad počtu položených otázok a časovej zložitosti programu nezabudnite zdôvodniť a dokázať.

Príklad

```

>>>3 2
0
>>>JOP
2 1 2
>>>NOP
2 1 3
>>>NOP
2 2 3
>>>JOP
-1
2 2 3

```

Pre lepšiu čitateľnosť sme pred riadky obsahujúce výstup testovača dopísali prompt „>>>“. Tento sa v skutočnosti nikde nezjaví: ani náš program ho nevypíše, ani vy ho nemáte vypisovať.

3. Dehumanizovaná klasifikácia

0 b za popis, 20 b za program

Jozef, ktorý už niekoľko týždňov pracuje ako vyšetrovateľ v organizácii PETA, sa nedávno zúčastnil jej každoročnej konferencie. Na tomto stretnutí sa do hĺbky skúmajú rôzne útoky na práva zvierat.

Jeho úloha spočíva v posúdení životného prostredia a etickosti zaobchádzania so zvieratami v zoologickej záhrade v Bratislave, ktorá sa nachádza veľmi blízko pri tajnej jadrovej elektrárni na matfyzu. Špecialisti v PETA zistili, že v tejto elektrárni je už viac ako 20 rokov prevádzkovaný softvér, ktorého autori disponujú len pochybnými zručnosťami v oblasti softvérového inžinierstva. Toto vyvolalo obavy, že elektráreň môže na zvieratá pôsobiť nadmernou radiáciou.

Jozef teda nahodil ponožky a sandále a vydal sa na špionážnu návštevu ZOO. Keďže vstupenka bola časovo obmedzená, musel pracovať rýchlo. Ešte pred návštevou vymyslel jeho tím geniálny plán ako posúdiť čo najviac zvierat za krátky čas. Každé zviera odfoťil špeciálnym fotoaparátom, ktorý využíva špičkové metódy kompresie. Teraz už len zostáva fotografie roztriediť podľa druhu zvieratá, ktoré na nich je. Na toto ale v PETA nikto nemá čas, keďže musia riešiť ďalšie naliehavé problémy a preto táto úloha zostáva na vás!

Úloha

Vytvorte program, ktorý načíta fotografiu z Jozefovho špeciálneho fotoaparátu a vypíše, aké je na nej zviera.

Formát vstupu

Na prvom riadku vstupu je číslo n ($1 \leq n \leq 4$), ktoré udáva počet riadkov obrázka. Nasleduje n riadkov obrázka, pričom každý má najviac 1000 znakov. Všetky riadky obrázka sú rovnako dlhé a prázdny priestor predstavujú medzery. Na načítanie riadku aj so všetkými medzerami odporúčame v jazyku C alebo C++ funkciu `getline(cin, premenna)` ktorá do premennej načíta riadok až po znak konca riadku.

Formát výstupu

Vypíšte na jeden riadok výstupu druh zvieratá na obrázku.

Príklad

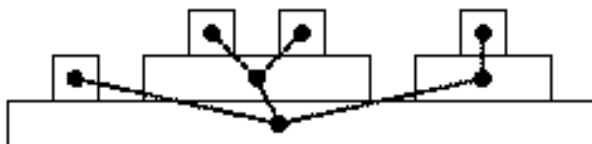
vstup	výstup
<pre>1 >0=====</pre>	<pre>had</pre>

4. i-Build

12 b za popis, 8 b za program

Denis sa v lete vybral na brigádu do továrne na kreatívne detské hračky, ktorú vlastní známa americká technologická firma. Zakladatelia tejto značky prišli s nápadom používať pri výrobe hračiek pre deti praktické a život zachraňujúce materiály. Napríklad, časti ich stavebnice sú vyrobené zo špeciálnej hmoty, ktorá pohlcuje radiáciu a tým chráni a lieči deti.

Stavebnica sa skladá z hranolov s rozmermi 1 cm , 1 cm a $n\text{ cm}$. Hranoly stavebnice sa ukladajú vždy len rovnobežne na seba, čím vytvárajú akési veže tak, ako je to znázornené na obrázku (pohľad z boku).



Platia pri tom tieto pravidlá:

- Vrch veže (hranol, na ktorom už nie je položený iný hranol) je kocka s rozmermi 1 cm , 1 cm a 1 cm .
- Hranol a položený na hranole b je kratší než hranol b .
- Medzi podstavami (spodkami) veží je medzera 1 cm .
- Medzi koncom hranola a podstavou veže je 1 cm .

Denis má k dispozícii len istý opis stavebnice. Každý hranol má svoje číslo x ($0 \leq x < n$), pričom n je počet hranolov v stavebnici. Denis dostane len zoznam $n - 1$ dvojíc hranolov a a b , ktoré sa dotýkajú (jeden je položený na druhom). To sa dá reprezentovať ako stromový graf ako na obrázku, pričom jeho koreň je v najdlhšom hranole a ten má vždy číslo 0.

Denisovou úlohou je presne zistiť, koľko centimetrov kubických hmoty je potrebných na výrobu konkrétnej stavebnice. Pomôžete mu?

Formát vstupu

Na prvom riadku vstupu je číslo t udávajúce počet stavebníc. Nasleduje popis t stavebníc. Na prvom riadku stavebnice je číslo n , ktoré hovorí o počte hranolov. Nasleduje $n - 1$ dvojíc hranolov a a b , ktoré znamenajú, že hranoly a a b sa dotýkajú.

Platí $1 \leq t \leq 500$ a $1 \leq$ súčet $n \leq 200\,000$.

Formát výstupu

Pre každú stavebnicu vypíšte jedno celé číslo udávajúce objem stavebnice v cm^3 (cm^3 nevypisujte).

Príklady

vstup

```
2
7
0 1
2 0
0 3
2 4
5 2
6 3
3
1 2
1 0
```

výstup

```
25
9
```

Na vstupe sú dve stavebnice. Prvá z nich je zobrazená aj na obrázku vyššie. Druhá z nich je jednoduchšia - je to pyramída výšky 3, kde najspodnejší hranol 0 má dĺžku 5, na ňom hranol 1 má dĺžku 3 a vrchný hranol 2 má dĺžku 1. To je dokopy objem 9cm^3 .

5. Áááale, to sa spraví...

8 b za popis, 12 b za program

V okolí reaktoru vraj nejak stúpala radiácia, a tak každý intern vyfasoval jeden Geigerov-Müllerov počítač a podho do terénu.

A ozať, ony tie merania vyšli nejak divne. Ale to je určite len nejaká fluktuácia chalani, do reportu to tak nepíšete. Trošku to upravte, tak žeby dobre bolo, a ono to určite samo časom prejde.

Úloha

Je a internov, každý spravil presne b meraní. Čísla a a b sú obe nepárne.

Medián všetkých meraní je hodnota, ktorá by bola uprostred, keby sme ich všetky usporiadali podľa veľkosti. Do reportu by sme potrebovali napísať také merania, ktorých medián bude **presne** m .

Každý intern môže klamať, a to tak, že prepíše nejaké svoje merania na ľubovoľné hodnoty. Koľko najmenej internov musí v reporte klamať?

(Každý intern musí do reportu napísať nejakých b čísel. Ak intern klame, nemusí nutne zmeniť všetky svoje hodnoty, môže upraviť len ľubovoľnú ich podmnožinu.)

Formát vstupu

V prvom riadku vstupu sú čísla a , b a m . V každom z nasledujúcich a riadkov je b čísel: merania spravené jedným z internov.

Všetky výsledky meraní aj číslo m sú celé čísla z rozsahu od 0 po 99. (Keď intern klame, môže do reportu uviesť aj iné čísla, ak chce.)

Hodnoty a a b sú kladné celé čísla. V jednotlivých sadách platia nasledovné obmedzenia pre ich súčin:

Sada	1	2	3	4
$\max a \times b$	7	99	999	99999

Formát výstupu

Vypíšte jeden riadok a v ňom jedno číslo: minimálny počet internov, ktorí musia v reporte klamať, ak potrebujeme, aby medián všetkých $a \times b$ meraní vyšiel presne m .

Príklady

vstup

```
5 5 8
1 2 3 4 5
10 9 8 7 6
25 24 23 22 21
18 16 17 19 20
11 13 12 14 15
```

výstup

1

Momentálne je medián všetkých meraní rovný 13. Na to, aby sa zmenšil na 8, napríklad stačí, ak štvrtý intern namiesto 18,16,17,19,20 dá do reportu 3,1,3,5,7.

vstup

```
5 5 8
1 2 25 24 23
3 4 22 21 20
5 6 19 18 17
7 16 15 14 13
8 12 11 10 9
```

výstup

2

Tu namerali interni tých istých 25 hodnôt, ale pri tom, ako ich majú teraz rozdelené, nestačí, aby klamal len jeden z nich.

vstup

```
1 5 30
30 50 10 20 40
```

výstup

0

vstup

```
1 5 27
30 50 10 20 40
```

výstup

1

vstup

```
3 1 20
10
10
10
```

výstup

2

6. Centrifúgy sa dotočia

12 b za popis, 8 b za program

V útrobach švajčiarskych hôr, na nevypátrateľnom mieste, prebieha každoročné zasadnutie Konzorcia Svetového Poriadku. Postavy zahalené v tme sedia po obvode kruhového stola a v miestnosti sa ozýva iba kovové chrapčanie modifikátorov hlasu. Svet čelí veľkým výzvam, imigrácii či globálnemu otepľovaniu, oni sú tu však na to, aby zabráňovali oveľa vážnejším, skrytým hrozbám.

Najnovšie správy od agentov v utajení uvádzajú rozmah nelegálnych centier spracovávajúcich rádioaktívny materiál. A je jasné, že takto získaný obohatený urán má len jediný cieľ. Je na čase, aby KSP zakročilo. Ich zásah musí však byť nenápadný. Nemôžu si predsa dovoliť, aby sa krajiny začali navzájom obviňovať a spôsobili ešte väčší chaos.

Ich riešenie je jednoduché. Vytvoria počítačový vírus, ktorý sa bude šíriť po sieti, či pomocou infikovaných USB kľúčov. Jeho úlohou bude vyhľadávať počítače, na ktoré sú napojené centrifúgy používané na oddelovanie rádioaktívneho materiálu a tieto centrifúgy zneškodňovať.

Centrifúga je pomerne jednoduché zariadenie. Po zapnutí začne postupne zvyšovať otáčky rotácie, až kým od kontrolného zariadenia nedostane signál na zastavenie. Aby sa zaručilo, že centrifúga bude spomaľovať plynule, signál na zastavenie pozostáva z viacerých znakov. Presnejšie, každá centrifúga má určené slovo S , ktoré sa používa na jej zastavenie. Kontrolný počítač vysiela centrifúge každú sekundu jedno písmeno a v prípade, že tieto písmená vytvoria slovo S , centrifúga zastane.

Cieľom navrhovaného vírusu by bolo zmeniť postupnosť písmen vyslaných kontrolným počítačom tak, aby centrifúga nezastavila, ale zrýchľovala tak dlho, až kým nedôjde k jej poškodeniu. Všetky vyslané signály sa však zaznamenávajú do logov, v záujme utajenia by preto bolo dobré, aby zmenené signály predsa len obsahovali slovo S . To však znie ako nemožná úloha – ako navrhnúť signál tak, aby obsahoval a zároveň neobsahoval slovo S ? Našťastie, agentom KSP sa podarilo získať zdrojové kódy centrifúg a nazdávajú sa, že časť overujúca signály je implementovaná zle.

Úloha

Na vstupe dostanete slovo S skladajúce sa z veľkých písmen anglickej abecedy. Vašou úlohou je vytvoriť **najkratší možný** (lebo nenápadnosť) reťazec X , ktorý obsahuje slovo S ako súvislý podúsek, ale nižšie uvedený program o ňom prehlási, že reťazec S neobsahuje.

K dispozícii máte ekvivalentné programy v jazykoch C++ aj Python. V oboch prípadoch je implementovaná funkcia `obsahuje(S, X)`, ktorá vracia `True` ak si myslí, že X obsahuje S a `False`, keď si myslí, že ho neobsahuje.

Listing programu (Python)

```
def obsahuje(S, X):
    ps, px = 0, 0
    while True:
        if ps == len(S):
            return True
        if px == len(X):
            return False
        if S[ps] == X[px]:
            ps += 1
            px += 1
        elif ps == 0:
            px += 1
        else:
            ps = 0
```

Listing programu (C++)

```
bool obsahuje(string S, string X) {
    int ps = 0, px = 0;
    while (true) {
        if (ps == S.length())
            return true;
        if (px == X.length())
            return false;
        if (S[ps] == X[px]) {
            ps += 1;
            px += 1;
        }
        else if (ps == 0) {
            px += 1;
        }
        else {
            ps = 0;
        }
    }
}
```

Formát vstupu

Na jedinom riadku vstupu sa nachádza slovo S tvorené n ($1 \leq n \leq 500\,000$) veľkými písmenami anglickej abecedy.

V polovici testovacích vstupov platí, že n je najväčšie 2 000.

Formát výstupu

Na výstup vypíšete do jedného riadku najkratšie možné slovo X , ktoré spĺňa požadované podmienky. V prípade, že takéto slovo neexistuje, na výstup vypíšete hlášku **Centrifuga sa nepokazi!**.

Príklady

vstup	výstup
AAB	AAAB
vstup	výstup
CENTRIFUGA	Centrifuga sa nepokazi!

7. Ideálne cestovanie metrom

12 b za popis, 8 b za program

V blízkej budúcnosti krajiny v rámci boja s globálnym otepľovaním zvýšili investície do hromadnej dopravy metrom (lebo je úspornejšia a hlavne v tomto teple pod zemou), a do jadrovej energie (lebo emisie).

Akonáhle všetky jadrové elektrárne povybuchovali, metro sa stalo najväčším centrom civilizácie¹. Obyvatelia metra však rýchlo objavili závažný problém – nie všetky tunely metra sú dostatočne hlboko pod zemou, aby cesta nimi bola zabezpečená voči radiácií z povrchu. Toto komplikuje cestovanie z jednej stanice metra do druhej, keďže priamy spoj môže byť viac zdraviu škodlivý ako ísť okľukou. Čo už, zabiť hodinu neoptimálnou trasou je lepšie ako zabiť seba otrávením radiáciou.

Artyom dostal o každom tuneli hlásenie o nameranej radiácií, a veľa dotazov od obyvateľov o najbezpečnejšej trase medzi dvoma stanicami metra. Radšej by však išiel s kamošmi loviť zmutované potkany ako sa prehrabávať mapami a radiť susedom ako najbezpečnejšie odnieť prádlo do pracovne. Poprosil vás preto o pomoc.

Úloha

Trasa medzi dvoma stanicami metra je postupnosť staníc taká, že prvá stanica je začiatková, posledná je konečná, a každá za sebou idúca dvojica staníc je spojená tunelom.

Nebezpečnosť cesty je najvyššie meranie radiácie v niektorom z tunelov, ktorý je na trase použitý.

Pre každú danú začiatkovú a konečnú stanicu metra nájdite najmenej nebezpečnú trasu medzi nimi, a vypíšete jej nebezpečnosť.

Formát vstupu

V prvom riadku sú čísla n a m – počet staníc metra a počet tunelov. V nasledujúcich m riadkoch sú trojice čísel a_i b_i c_i , znázorňujúce tunel medzi stanicami a_i a b_i s výškou radiácie c_i . Stanice sú očíslované od 1 po n , a je zaručené že z každej stanice sa dá dostať do každej inej nejakou postupnosťou tunelov.

Za nimi sa nachádza číslo q – počet dotazov. Každé z ďalších q riadkov obsahuje dve čísla z_i k_i , dotaz na najbezpečnejšiu cestu začínajúcu na stanici z_i a končiacou stanicou k_i .

Vo všetkých vstupoch platí $1 \leq n, q$ (aspoň niečo budete mať robiť), $n - 1 \leq m \leq \frac{n(n-1)}{2}$ (ciest je dosť na prepojenie metra, ale nie viac ako možný počet priamych spojení), $a_i \neq b_i$ (tunel nevedie zo stanice naspäť na ňu) $1 \leq c_i \leq 10^9$.

Sada	1	2	3	4
n	15	2 033	100 000	300 000
m	15	2 033	100 000	300 000
q	15	50 000	100 000	300 000

¹nemalo na povrchu príliš konkurenciu

Formát výstupu

Pre každý dotaz vypíšte najmenšie číslo $0 \leq r_i$ také, že sa dá dostať zo stanice z_i do k_i a pritom nepoužiť tunel s radiáciou vyššou ako r_i .

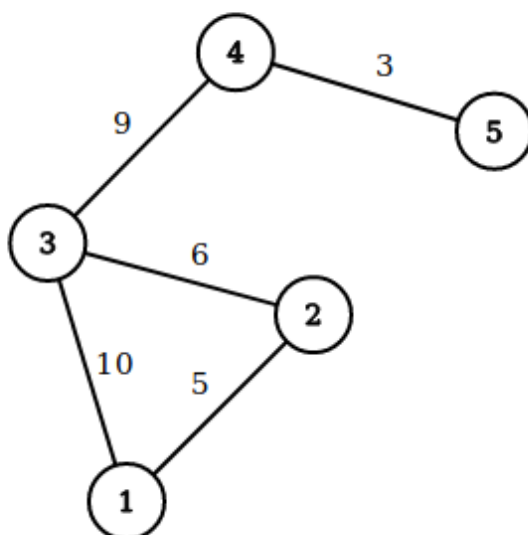
Príklad

vstup

```
5 5
1 2 5
3 1 10
2 3 6
3 4 9
4 5 3
5
3 1
3 4
3 5
4 5
5 5
```

výstup

```
6
9
9
3
0
```



Takto vyzerá sieť metra z príkladu. V prvom dotaze sa nám oplatí ísť okľukou cez stanicu 2.

8. Anihilácia

12 b za popis, 8 b za program

Z reaktora sme vybrali trubice s rádioaktívnym odpadom, ktorý potrebujeme ekologicky zlikvidovať. V každej trubici sú zaradom kusy rôznorodých rádioaktívnych látok, pre jednoduchosť ich budeme označovať písmenami a až z. Ak sa v trubici nachádza za sebou kritické množstvo (dva alebo viac kusov) rovnakej látky, môžeme na tento úsek trubice vystreliť prúd neutrónov a celý tento úsek tým zničiť. Látky v trubici, ktoré boli pred tým na koncoch zničeného úseku budú po tomto vedľa seba.

Ak máme napríklad v trubici látky `aabcccbacc` môžeme najprv zničiť úsek štyroch látok `c` a v trubici budú látky `aabbacc`. Potom môžeme zničiť úsek dvoch látok `b`, následne troch látok `a` a napokon zničíme zvyšné látky `c`.

Týmto postupom by sme zničili celý obsah trubice, avšak ak by sme napríklad najprv zničili úsek látok `a`, už by sme celý obsah trubice nevedeli zlikvidovať.

Úloha

Máte zadaných niekoľko reťazcov malých písmen anglickej abecedy. Zistite, pre ktoré reťazce dokážeme

opakovaným odstraňovaním súvislých úsekov dvoch alebo viacerých rovnakých písmen odstrániť celý reťazec.

Formát vstupu

Na prvom riadku vstupu je prirodzené číslo n ($1 \leq n \leq 20$) udávajúce počet testovacích sád. Nasleduje n riadkov predstavujúcich jednotlivé testovacie sady, na každom riadku je jeden reťazec písmen a až z. Dĺžka každého reťazca je aspoň 1 a najviac 200 znakov.

Presnejšie, v jednotlivých vstupoch (1.in až 8.in) platí, že dĺžka každého reťazca je zhora obmedzená postupne 5, 10, 20, 40, 80, 160, 200 a 200.

Formát výstupu

Pre každú testovaciu vypíšte jeden riadok obsahujúci slovo **ano** ak sa dá celý reťazec odstrániť a **nie**, ak sa nedá.

Príklady

vstup

```
4
aabcccbacc
bcccbacc
hahaahh
xzzxyzzzyzx
```

výstup

```
ano
nie
nie
ano
```