Problem A. Бағдаршам

Жауаптар келесі форматта: $input \to output$. $jasyl \to 0$, $sary \to 1$, $qyzyl \to 2$.

Problem B. Жаңа графтағы қысқаша жолдар

Дейкстра алгоритмын қолданамыз. Бірақ жаңа құрылған графта қырлар саны өте көп болуы мүмкін. Бізге тек мәні ең кішкентай таңдалмаған төбены алып отыру қажет.

Әр төбе үшін оның бастапқы графтағы әлі таңдалмаған қырлардың жиының сақтап отырамыз. Ең кіші мәні салмағы ең кіші қырда болатының байқаймыз. Бұл қырдың мәнің есептеу үшін бізге деректер құрылымы қажет. Содан мәні ең кіші қырды табу үшін, бізге n мүмкін мәннің ең кішкентайын таңдау қажет.

Дейкстра алгоритмында қырды қарастырғанда, осы қырдың соңдары үшін: бұл қырды таңдалмаған қырлар жиынынан алып тастаймыз, қырды деректер құрылымына қосамыз және жаңа ең кіші қыр үшін оның мәнің есептейміз.

Нэтижесінде біздің деректер құрылымымыз келесі әрекеттерді қолдауға қабілетті болуы керек:

1. Берілген x үшін $x \cdot w_e + dp_e$ мәндер арасындағы ең кіші мәнді табу.
2. Жаңа (dp_e, w_e) жазбасын қосу.

Мұны біз *online convex hull trick* немесе *li chao tree* алгоритмдері арқылы іске асыра аламыз.

Егер dp_e өсу ретімен қосылатының ескерсек, онда *convex hull trick*-тың (k_i, b_i) жұптары b_i өсу ретімен қосылатын нұсқасын қолдансада болады. Жауапты бинарлық ізденіс немесе екі нұсқағыш арқылы тапсада болады, себебі xте өсу ретімен сұралады.

Уақыт күрделігі $O(m \cdot \log(m))$.

Problem C. Жиымды қалпына келтіру

Әр сан 1 мен N арасында болғандықтан, онда N санына тек N саны бөлінеді. Олай болса B_N дегеніміз A массивінде N-ға тең элементтердің саны. Оларды алып тастап, жаңа B_i мәндерін санайтын болсақ, онда қалған сандар $N-1$-ден аспайды. Осы процесті N рет қайталаймыз.

python тіліндегі код:

```
n = int(input())
b = list(map(int, input().split()))
ans = []

for i in range(n, 0, -1):
    for j in range(1, b[i - 1] + 1):
        ans.append(i)
    for j in range(1, i):
        if i % j == 0:
            b[j - 1] -= b[i - 1]
print(*ans)
```

Problem D. Жиымдағы сұраулар туралы кезекті бір есеп

Жиымды бірдей сандардың блоктары ретінде сақтайық. Мысалы, $[1, 1, 0, 0, 0, 3, 3, 1]$ жиымы келесідей үштіктер жиымымен сақталатын болады: $[(1, 2, 1), (3, 5, 0), (6, 7, 3), (8, 8, 1)]$. Мында әрбір (l, r, x) үштігі барлық i $(l \leq i \leq r)$ үшін $a[i] = x$ екенін білдіреді.

Әр сұрау кейбір үштіктерді жойып, үштен көп емес жаңа үштік қоса алады. Жаңа үштіктердің біреуі аралықтың өзі болады. Екі қосымша үштік біздің аралық блоктарды ортасынан «кесіп» жатқан болса қосыла алады.

Аралықтағы сандардың MEX мәнін табу үшін, сол аралықтағы бүкіл блоктарды тауып, ең бірінші кездеспейтін санды табуға болады. Бұл аралықтағы блоктар саны уақытында жұмыс істейді. Бір қарағанда, бұл тез $O(n^2)$ болып кетеді.

Дегенмен әрбір блок тек бір рет қана қаралады. Жалпы блоктардың саны $n + 3q$-тен аспайтынын білеміз. Сондықтан қорытынды күрделілік $O(n + q)$ болады.

Блоктарды сақтау үшін *set* немесе *segment tree* деректер жиымын қолдануға болады. Соңында уақыт күрделілігі $O((n + q) \log n)$ болады.

Problem E. Киіз үй

Егер ауылдың орталығын P нүктесіне қойсақ, барлық киіз үйлерді 2 түрге бөлуге болады: P нүктесіне қатысты симметриялы жұбы барлар және жоқтар. Әрбір жұбы жоқ киіз үй үшін жауапқа $+1$ қосуымыз керек.

Керісінше, әрбір P нүктесі үшін, қанша киіз үйге жауапты үлкейту **керек жоқ** болатынын санайық. Бұл мәнді $M[P]$ деп атайық. Демек бізге $n - M[P]$ жаңа киіз үй қосу керек болады.

Барлық киіз үйлердің A, B жұптарын қарап шығайық. Бұл жұп $P = (A + B)/2$ нүктесіне қатысты симметриялы болады. Демек біз $M[(A+B)/2]$ мәніне 2 қоса аламыз. Сонымен қатар, әрбір A нүктесі үшін $M[A]$ мәнін 1-ге көбейтейік (өйткені әр нүкте өз-өзіне қатысты әрдайым симметриялы болады).

Енді жауап $n - \max_A M[A]$ мәніне тең болады. M мәндерін *map* деректер жинағында оңай сақтасақ болады. Қорытынды уақыт күрделілігі: $O(n^2 \log(n))$.

Problem F. Арман және құрылыс ойыны

$n \leq m$ деп есептейік. Керісінше болса, бір әрдайым кестені 90 градусқа бұра аламыз.

2-ден басқа әрбір жай сан тақ болатынын байқайық. 3 және $n \times m$ арасындағы барлық жай сандарды санайық. Бұл мәнді k деп белгілейік.

Егер $k = 0$ болса, кез-келген кестені шығара аламыз. Бұл мезеттен бастап $k > 0$ деп есептейміз.

Егер $n = 1$ болса, бізде барлық жағдайда сомасы тақ болатын бағандар болады. Сондықтан осында жауап жоқ.

Егер $k = 2$ болса, сол екі жай санды бір жолға қоюға болады. Бірақ ол кезде бағадардың сомасы тақ болады. Керісінше де дәл солай. Сондықтан осында да жауап жоқ.

Егер k тақ сан болса, жауап осында да жоқ. Өтйкені әрбір жолдың сомасы жұп болу керек болса, демек бүкіл кестенің де сомасы жұп болу керек.

Егер k қалдықсыз 4-ке бөлінсе, кестеде 2×2 шаршыларын қойып шыға аламыз. Әрбір осындай шаршы түк өзгертпейтіндіктен, бұл әрдайым жұмыс істейді.

Алайда, бізге керек шаршылардың саны кестеге сыймай жатса не істейміз? Мұндай жағдай ешқашан болмайтындығына сенуге болады, өйткені $k - n \times m$-ге дейінгі **жай сандардың** саны. Ал жай сандар бүкіл кестенің шаршыларымен салыстырғанда аса көп емес (N мәніне дейінгі жай сандардың санын $\frac{N}{\log N}$ деп пайымдауға болады).

Ал егер k саны 4-ке бөлінбесе, бірақ 2-ге бөлінсе ше? Мысалы $k = 6$? Осындай кезде біз ұқсас түрде келесідей 3×3 шаршысын құра аламыз:

```
XX.
.XX
X.X
```

Мында, «X» жай сан қойылған ұяшықты білдіреді. Бұл $k = 6$ жағдайына жауап болатынына оңай көз жеткізуге болады. Осы жерден бізде тағы бір жағдай шығады.

Егер $n = 2$ болса, жауап жоқ, өйткені 3×3 шаршысы кестемізге сыймайды.

Қалған барлық жағдайда жауап әрдайым бар болады. Алдымен 3×3 шаршысын кез-келген бұрышқа қояйық және k мәнінен 6-ны азайтайық. Енді k нақты 4-ке қалдықсыз бөлінетін болады, ал бұл жағдайды біз шығара аламыз.

Қорытынды уақыт күрделілігі: $O(\sum nm)$.

Problem G. Қызыл-қара дарақ

Белгіленген бояудың жауабын есептейік.

Дарақтың түбірі ретінде 1 төбесін алайық. Енді cnt_v — v төбесінің кіші дарағындағы(subtree) қызыл төбелер саны. Онда жауап $\sum \min(cnt_i, k-cnt_i)$ тең болады. Мұны әр қыр арқылы неше жұп өтетінің қарау арқылы дәлелдеуге болады.

Бір төбенің түсін ауыстырғанда k және түбірге дейінгі төбелердің cnt_i өзгереді.

Әр төбе үшін $dp_{v,x,y}$ — оның жауапқа қосатын мәнің есептейік, егер дарақтағы қызыл төбелер саны x және $cnt_v = y$ болса.

Онда v төбесінің түсін ауыстырғанда:

- Егер бастапқыда v қызыл түсті болса, онда жауап келесі мәндердің қосындысы:
  - $dp_{u,k-1,cnt_i-1}$ түбірден v төбесіне дейінгі жолдағы барлық u үшін.
  - $dp_{u,k-1,cnt_i}$ қалған барлық төбелер үшін.
- Егер бастапқыда v қызыл түсті болса, онда жауап келесі мәндердің қосындысы:
  - $dp_{u,k+1,cnt_i+1}$ түбірден v төбесіне дейінгі жолдағы барлық u үшін.
  - $dp_{u,k+1,cnt_i}$ қалған барлық төбелер үшін.

Бұл дегеніміз dp-ның тек $O(n)$ мәнің санап, олардың түбірге дейінгі соммасын сақтап білу қажет.