Задатак 1: Фибоначијеви стрингови

Низ Фибоначијевих стрингова се рекурзивно дефинише на следећи начин:

$$F(n) = \begin{cases} 0, & \text{ако је } n = 0, \\ 1, & \text{ако је } n = 1, \\ F(n-1) + F(n-2), & \text{ако је } n \geq 2. \end{cases}$$

Притом, оператор + означава конкатенацију стрингова. Првих неколико елемената низа Фибоначијевих стрингова су следећег облика:

$$\begin{aligned} F(0) &= 0, \\ F(1) &= 1, \\ F(2) &= 10, \\ F(3) &= 101, \\ F(4) &= 10110, \\ F(5) &= 10110101. \end{aligned}$$

За дати низ битова p и индекс n одредити колико пута се стринг p јавља у стрингу $F(n)$.

У првој линији стандардног улаза налази се природан број n, $1 \leq n \leq 100$. Друга линија улаза садржи непразан стринг p, који може имати највише 100 000 карактера. На стандардни излаз исписати број тражених појављивања. Појављивања стринга p се могу преклапати. Гарантује се да број појављивања неће ни у једном тест примеру бити већи од 2^{63}.

Пример улаза:
7
10

Пример излаза:
8

Решење. Основни проблем са овим задатком је што дужина Фибоначијевих стрингова расте експоненцијално, па није могуће непосредно проверити за велике n колико је пута p подстринг стринга $F(n)$. Имајући у виду да је $F(n) = F(n-1) + F(n-2)$, а број битова у p није већи од 100 000, за фиксирано n, могу се непосредно одредити сва појављивања стринга p која захватају границу спајања стрингова $F(n-1)$ и $F(n-2)$, а затим на ту вредност додати већ израчунате вредности за случајеве $n-1$ и $n-2$. Првих неколико вредности се могу израчунати непосредно. Због ефикасности, за одређивање колико пута се подстринг p налази у датом стрингу, може се користити Кнут-Морис-Пратов алгоритам (енгл. *Knuth-Morris-Pratt*).

Задатак 2: Факторизација

Према основној теореми алгебре, сваки природан број већи од 1 се може на јединствен начин раставити на просте факторе. Притом, имајући у виду комутативност, он се на више начина може представити као њихов производ. Тако је, на пример, $6 = 2 \cdot 3 = 3 \cdot 2$ и $18 = 2 \cdot 3 \cdot 3 = 3 \cdot 2 \cdot 3 = 3 \cdot 3 \cdot 2$. Нека је $f(k)$ број различитих представљања броја k. Тако је $f(6) = 2$ и $f(18) = 3$. За сваки позитиван цео број n постоји бар један број k за који је $f(k) = n$. Наћи најмањи такав индекс k.

У првој линији стандардног улаза се налази природан број n, $n \leq 2^{63}$. На стандардни излаз је потребно у једној линији исписати тражени број k. Гарантује се да је $k \leq 2^{63}$.

Пример улаза:
3

Пример излаза:
12

Решење. Нека је растављање броја k на просте чиниоце дато са $p_1^{t_1} p_2^{t_2} \cdots p_r^{t_r}$. Имајући у виду комутативност, број различитих представљања броја k као производ простих фактора је

$$\binom{t_1 + t_2 + \cdots + t_r}{t_1, t_2, \ldots, t_r} = \frac{(t_1 + t_2 + \cdots + t_r)!}{t_1! t_2! \cdots t_r!}.$$

Без умањења општости можемо претпоставити да је $t_1 \geq t_2 \geq \cdots \geq t_r$. На овај начин се могу генерисати сви бројеви облика $k = 2^{t_1} 3^{t_2} 5^{t_3} \cdots$ за које је $t_1 \geq t_2 \geq t_3 \geq \ldots$ и $1 < k < 2^{63}$. Таквих бројева има мање од 50 000. За сваки такав број можемо израчунати вредност $f(k)$ и проверити да ли је мања од 2^{63}, а затим конструисати и сачувати вредности за свако такво $f(k)$ (испоставља се да таквих бројева има мање од 20 000).

Задатак 3: Градови

Дато је n ($1 \leq n \leq 50\,000$) градова који су повезани са m ($1 \leq m \leq 100\,000$) двосмерних путева. Пут i повезује градове a_i и b_i, ако је испуњено $a_i \neq b_i$. Притом два различита пута могу повезивати два иста града. Градове је потребно означити битовима 0 и 1, тако да никоја два града која су повезана неким путем немају исту ознаку. Како је цена означавања града помоћу 1 скупља од цене означавања неког града помоћу 0, потребно је максимизовати број градова који су означени са 0.

Са стандардног улаза се у првом реду учитавају бројеви n и m. У наредних m редова се учитавају вредности градова a_i и b_i ($1 \leq a_i, b_i \leq n$), што означава да је град a_i повезан градом b_i помоћу i-тог пута. На стандардни излаз је потребно исписати максималан број битова 0 којим се могу означити градови.

Пример улаза:
4 4
1 2
2 3
3 4
4 1

Пример излаза:
2

Решење. Мрежа путева се може представити графом чији су чворови градови, а ивице путеви. Граф је најпре потребно поделити на повезане компоненте, а онда за сваку повезану компоненту проверити да ли се може обојити са две боје. За сваку повезану компоненту, довољно је кренути од једног чвора и обојити га нпр. битом 0. Затим, користећи претрагу по дубини, редом бојити чворове, водећи рачуна да никоја два суседна чвора нису обојена истом бојом. Уколико је немогуће извести такво бојење на било којој повезаној компоненти, исписати -1. Уколико је могуће обојити, треба проверити број 0 и број 1 у градовима. Ако је број 1 већи од броја 0, потребно је инвертовати одговарајуће битове. Уколико ниједна повезана компонента нема резултат -1, тада је коначан резултат једнак збиру броја 0 за сваку компоненту.

Задатак 4: Куглице

Милан има n $(1 \leq n \leq 10\,000)$ гомила са куглицама, при чему се на свакој гомили налази једнак број куглица. Његов млађи брат Јован воли да се игра са његовим куглицама, тако што произвољан број куглица премести из једне кутије у другу и тај процес понови неколико пута. Као резултат, добија се низ кутија који не морају имати увек једнак број куглица. Како Милан воли да се у свим његовим кутијама налази једнак број куглица, он жели да што пре кутије врати у првобитно стање. Помозите Милану да у што мањем броју корака испремешта куглице, тако да се у свакој кутији налази једнак број.

У првој линији улаза се налази укупан број кутија n, а у наредних n редова се налази број куглица у свакој кутији, који представља ненегативан цео број који није већи од 10 000. На стандардни излаз је потребно исписати тражени минимални број премештања.

Пример улаза:
4
2
10
7
1

Пример излаза:
7

Објашњење примера: Након померања 3 куглице из друге у прву кутију, 2 куглице из друге у четврту и 2 куглице из треће у четврту кутију, у свакој кутији ће се налазити по 5 куглица.

Решење. Просечан број куглица је једнак количнику броја свих куглица и броја кутија. На крају је довољно одредити збир оних куглица из сваке кутије за које је полазни број куглица већи од добијеног просека. За сваку такву кутију, на збир се додаје број куглица у кутији, а затим одузима просечан број куглица. Укупан збир таквих куглица ће представљати решење. Приметимо да није потребно да експлицитно одређујемо на који начин се куглице морају преместити.

Задатак 5: Премештање

Дат је низ од n елемената, који представља неку пермутацију првих n природних бројева. Над низом се могу задавати две врсте упита:

- $1\ i\ j$ – елементи низа на позицијама i и j морају заменити места;
- $2\ i\ j$ – проверава се да ли су елементи низа чије су вредности $i, i+1, \ldots, j$ на узастопним местима у низу.

У првој линији стандардног улаза се налазе бројеви n и m ($2 \leq n, m \leq 200\ 000$), који редом представљају број елемената низа и број упита. У другом реду се налази n природних бројева, чије су вредности од 1 до n, при чему се свака вредност појављује тачно једном. У наредних m редова се налазе упити облика $1\ i\ j$ ($1 \leq i, j \leq n$, $i \neq j$) и $2\ i\ j$ ($1 \leq i \leq j \leq n$). На стандардни излаз је потребно за сваки упит типа $2\ i\ j$ исписати великим словима енглеске абецеде DA или NE, у зависности од тога да ли су елементи низа чије су вредности $i, i+1, \ldots, j$ на узастопним местима у низу.

Пример улаза:
5 3
2 4 1 3 5
2 2 5
1 3 1
2 2 5

Пример излаза:
NE
DA

Решење. Нека t_k представља тренутну позицију елемента k у низу. За операцију облика $1\ i\ j$, довољно је заменити вредности елемената низа t_i и t_j. За операцију облика $2\ i\ j$, довољно је пронаћи најлевљу и најдешњу позицију међу свим бројевима од i до j и проверити да ли су они удаљени за тачно $j - i$ позиција, јер у том случају неће бити других елемената између. Дакле, за $M = \max\{t_k : i \leq k \leq j\}$ и $m = \min\{t_k : i \leq k \leq j\}$, потребно је проверити да ли је $M - m = j - i$. Уколико јесте, елементи $i, i+1, \ldots, j$ су на узастопним местима у низу, а иначе нису. Да би се ова операција ефикасно извела довољно је користити интервално стабло (енгл. *interval tree*), где у сваком чвору можемо памтити максимум и минимум на одговарајућем интервалу.

Задатак 6: Медијана

Медијана низа са непарним бројем елемената представља онај елемент који има средњи индекс у низу који се добија сортирањем полазног. На пример, за низ $5, 1, 17, 10^7, 999$, медијана је 17, а за низ $1, 1, 1, 99, 2$, медијана је 1.

Са стандардног улаза се у првом реду учитава број n, $1 \leq n \leq 100\,000$. У наредних n редова се редом учитавају елементи низа, који представљају реалне бројеве заокружене на три децимале. На стандардни излаз је потребно у сваком тренутку када је број тренутних учитаних елемената непаран, исписати која је медијана тренутног низа.

Пример улаза:
5
17
1
1
3
5

Пример излаза:
17
1
3

Решење. Задатак се може ефикасно решити користећи бинарни хип (енгл. *binary heap*). Нека је h_{max} бинарни хип у коме је вредност коренског чвора максимална, а h_{min} бинарни хип у коме је вредност коренског чвора минимална. При операцији додавања наредног елемента разликују се два случаја: први, код кога је тренутна позиција p елемента непарна; и други, код кога је тренутна позиција p датог елемента парна. Ако је тренутна позиција p елемента у низу непарна, тада се у хипу h_{max} налази половина тренутних елемената који имају мању вредност, а у хипу h_{min} налази половина тренутних елемената који имају већу вредност. Дакле, највећа вредност у хипу h_{max} је мања или једнака од најмање вредности у хипу h_{min}. Нови елемент се најпре убаци у хип h_{max} на одговарајуће место. Ако сада корен хипа h_{max} и даље није већи од корена хипа h_{min}, једноставно се испише корен h_{max}. Ако то није случај, потребно је први елемент из хипа h_{max} убацити на одговарајуће место у хип h_{min}, и обратно, а затим исписати коренски елемент хипа h_{max}. Када је вредност p парна, елемент се најпре додаје у хип h_{min}, па се, аналогно претходном, по потреби замене коренски чворови два хипа. Број елемената два хипа се тиме увек разликује највише за 1, а операције додавања и брисања из хипа се могу извести у времену $O(n)$, где је n број тренутних елемената у хипу.

Задатак 7: Стаза

Стаза по којој се креће пешак има облик два спојена једнакокрака троугла. У горњој половини стазе се у првом реду налази један ненегативан цео број, у другом два, ..., а у n-том n бројева. У доњој половини стазе се у $(n+1)$-ом реду налази $n-1$ бројева, у $(n+2)$-ом $n-2$, ..., а у последњем реду један број. На пример, једна стаза за коју је $n=4$ може бити следећег облика:

5
2 7
9 3 1
8 4 2 6
4 5 5
3 0
4

У сваком тренутку, пешак од тренутне позиције може, уколико је то могуће, скренути лево или десно надоле. Он најпре креће од броја у првом реду, пролази кроз по један број у сваком реду, а зауставља се на броју који се налази у последњем реду. Одредити такву путању пешака да бројеви на које он наиђе у збиру дају минималну суму.

Са улаза се у првом реду најпре учитава број n ($n \leq 5000$). У другом реду се најпре учитава један број, у трећем два, ..., а у $(n+1)$-ом n бројева. Након тога се у $(n+2)$-ом реду учитава $n-1$ бројева, у $(n+3)$-ем $n-2$, а у последњем $(2n+1)$-ом реду један број. Бројеви на улазу су поравнати налево, док су у примеру задатка центрирани. На стандардни излаз је потребно исписати један цео број, који представља минималну тражену суму.

Пример улаза:
3
1
7 4
10 1 5
17 12
9

Пример излаза:
27

Решење. Задатак се може једноставно решити динамичким програмирањем. Нека је A улазна матрица димензија $(2n-1) \times n$, где су у сваком реду елементи до n-те колоне попуњени нулама. Нека је матрица B на почетку једнака матрици A, тј. нека је $B_{ij} = A_{ij}$, $1 \leq i \leq 2n-1$, $1 \leq j \leq n$. За $i > 1$ и $j > 1$, биће $B_{ij} = B_{ij} + \min\{B_{i-1,j-1}, B_{i-1,j}\}$. Решење је вредност $B_{2n-1,1}$.

Задатак 8: Огрлица

Задата је огрлица састављена од плавих и црвених куглица. Куглице су обележене бројевима од 0 до $n-1$. Ако прекинемо везу између две суседне куглице, од огрлице можемо да добијемо низ куглица. На пример, прекидом везе између треће и четврте куглице код огрлице

```
       11 0 1
        B B B
   10 R     R 2
   9 R       R 3
     8 B     R 4
        B R R
        7 6 5
```

добија се низ

```
4 5 6 7 8 9 10 11 0 1 2 3
R R R B B R  R  B B B R R
X X X               X X
```

Овде су са B означене плаве, а са R црвене куглице. Куглице се након прекида прикупљају на следећи начин. Са десне стране, узимамо једну куглицу (у овом случају црвену) и идемо налево док не наиђемо на куглицу супротне боје (плаву). Исти поступак применимо и са леве стране. Добили смо пет куглица. Треба наћи такво место прекида, да је број прикупљених куглица максималан (иста куглица се не може прикупити два пута). У овом случају је то веза између прве и друге куглице. Добијамо низ и прикупљамо осам куглица:

```
2 3 4 5 6 7 8 9 10 11 0 1
R R R R B B R  R  B B B
X X X X X          X X X
```

Са стандардног улаза се учитава низ куглица у облику малих слова r и b енглеске абецеде, која представља конфигурацију почев од нулте куглице. Максималан задати број куглица је 5000. За горњи пример линија би гласила: *bbrrrrrbbrrb*. На стандардни излаз је потребно исписати место прекида где је број прикупљених куглица максималан, као и број прикупљених куглица. Ако има више таквих места, исписује се оно са највећом позицијом прекида. У горњем примеру место прекида је између друге и треће куглице, а број прикупљених куглица је 8. Претпоставити да огрлица неће бити једнобојна.

Пример улаза:
rrbbbbrrrbbbb

Пример излаза:
8 9
7

Решење. Редом испитујемо колико куглица можемо добити прекидом огрлице између сваке две узастопне куглице: n и 1, 1 и 2, ..., $n-1$ и n. Бележимо максималан број куглица и између којих куглица се такво максимално прикупљање добија. Испитивање куглица $i, i+1$ вршимо бројањем колико од куглица $i-1, i-2, \ldots$ (куглица после прве је n-та) је исте боје као куглица i. Слично, пребројимо колико куглица $i+2, i+3, \ldots$ је исте боје као куглица $i+1$.

Задатак 9: Мапа

Дата је мапа града који се састоји од m улица у смеру исток-запад и n улица у смеру север-југ ($1 \leq m, n \leq 200$). Сваке две улице у граду су на мапи или паралелне или су секу по тачно једној раскрсници. Надморске висине раскрсница су различите и налазе се у матрици X, где X_{ij} представља надморску висину раскрснице у којој се секу i-та улица правца исток-запад и j-та улица правца север-југ. Део било које улице између сваке две суседне раскрснице је дуж. За две дате раскрснице раскрснице A и B, испитати да ли постоји пут улицама тог града, којим се може стићи из A у B тако да се може кретати стално низбрдо (дакле, дозвољено је кретати се само од раскрснице са већом у суседну раскрсницу са мањом надморском висином). Потребно је одредити у колико најмање корака се може стићи из A у B.

Са улаза се у првом реду учитавају бројеви m и n. У наредна два реда се учитавају координате града A и града B. У последњих m редова се у сваком реду учитава n бројева, који представљају елементе матрице X. На стандардни излаз је потребно исписати тражени минимални број корака. Уколико није могуће стићи из A у B, исписати -1.

Пример улаза:
3 3
0 0
2 2
9 8 7
6 5 4
3 2 1

Пример излаза:
4

Решење. Идеја овог задатка је да се формира матрица P, где је $P_{ij} = k$, ако постоји пут из полазног места до (i, j)-те раскрснице крећући се само низбрдо. Сви елементи се на почетку иницијализују са -1, а почетна раскрсница са 0. За сваку текућу раскрсницу у сваком кораку проверавамо да ли се до суседне из полазне може доћи низбрдо и да ли је суседна виша од текуће. Ако су оба услова испуњена, вредност у текућој раскрсници се поставља за 1 већу од суседне. Програм се завршава у случају када стигнемо до одредишне тачке или када више не можемо да се крећемо низбрдо.

Задатак 10: Израз

Дат је израз који садржи операције сабирања и множења. Време потребно да се изврши сабирање је p, а време потребно да се изврши множење је q. Време потребно да се изврши операција $A * B$ ($*$ је множење или сабирање) једнако је времену потребном да се изврши операција $*$, плус дужем од времена потребних за израчунавање подизраза A, односно B (подизрази A и B се извршавају паралелно). Операнди су променљиве које се састоје од једног карактера, а време потребно за њихово израчунавање је 0. Написати програм који са стандардног улаза чита вредности за p и q, а затим и израз (израз садрзи највише 1000 карактера, без белина). Заграде се обавезно користе да одреде редослед извравања операција. За сваки израз наћи и приказати време потребно за његово израчунавање. Претпоставити да су изрази коректно задати.

Са стандардног улаза се у првом реду учитавају вредности p и q, а у другом реду дати израз. На стандардни излаз је потребно исписати тражено минимално време извршавања.

Пример улаза:
4 6
$((((a * b) * c) * d) + e) + (f * g)$

Пример излаза:
26

Решење. При процесуирању датог израза, потребно је пронаћи најлевљи знак множења или сабирања који се налази ван заграда. Ово се једноставно може урадити уколико се води рачуна о тренутном броју отворених и затворених заграда, крећући се кроз израз слева надесно. Вредност таквог израза је једнака вредности операције, на коју је додата максимална вредност од два израза – оног који је лево од знака операције и оног који је десно од знака операције. Затим се леви и десни подизраз могу посматрати рекурзивно. На почетку процесуирања израза, у сваком кораку, потребно је водити рачуна о сувишним заградама слева и здесна.