Назив проблема: Сплит

|  |  |
| --- | --- |
| Аутор: Петар Величковић | Анализа: Петар Величковић |

|  |  |
| --- | --- |
| Тагови: | кумулативне суме, структуре података, Фенвикова стабла |

## Решење и анализа:

Проблем Сплит је један од тежих проблема на другим Квалификацијама, и да би се потпуно решио потребно је знање напреднијих структура података (ниво Српске Информатичке Олимпијаде). Инспирација за задатак је потекла од задатка Жетони са СИО 2012. када је аутор (тадашњи такмичар) употребио сличну методу ”сплитова” да би решио задатак.

Задатак се може превести у формални облик на више начина; ево једне могуће интерпретације:

*Задата је функција са скупа природних бројева на скуп стрингова, , тако да у почетку важи за све . Сплит операцијом на позицији са стринговима u функција се мења у функцију тако да важи:*

*ако .*

*ако .*

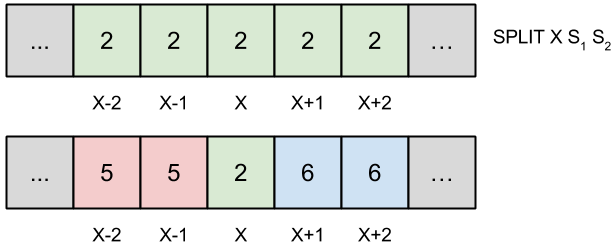
*иначе.*

*Потребно је одредити сваки пут када се позове query операција на позицији n.*

Задатак се може решити тако што одржавамо структуру **интервала**, где за сваки интервал памтимо неки јединствени идентификатор, леву и десну границу, као и стринг који означава програм који се тренутно извршава на том интервалу. У почетку постоји само један интервал, , тако да важи: . Неопходно је памтити који је последњи направљени интервал.

У исто време треба одржавати и низ , тако да значи да се поље тренутно налази унутар интервала . Наравно, у почетку је за све . Одржавањем овог низа добијамо једноставан одговор на све упите; довољно је вратити .

Посматрајмо сада како одржавати током неке сплит операције:

На слици десно се налази низ , пре и после сплит операције на пољу . Поља у интервалу су под контролом интервала , а последњи до сада направљени интервал је .

Сплит операција креира два нова интервала, и , и додељује им следеће параметре:

;

.

Такође се низ ажурира као на слици, и .

Наивно решење које одржава низ тако што при свакој сплит операцији преписује све његове чланове обухваћене неким интервалом није довољно ефикасно и доноси ~50 бодова. Да бисмо закорачили ка комплетном решењу не смемо експлицитно чувати овај низ у меморији; уместо тога одржаваћемо низ , где ће бити сума свих елемената низа од првог до -тог; формалније:

Очигледно је да у почетку за све . Приметимо да, уколико желимо да на неком интервалу повећамо све вредности низа за , довољно је урадити:

Назваћемо ову методу . Могуће је извршити сплит операцију са константним бројем позивања ове методе; aко претпоставимо да желимо извршити сплит операцију на пољу , да је интервал који обухвата , и да је последњи направљени интервал до сад, сплит операција се може описати следећим алгоритмом:

1. Направити интервале и као што је раније описано.
2. Ажурирати тако да за све .
3. Aжурирати тако да за све .
4. Aжурирати тако да за све .
5. Aжурирати тако да за .
6. Ажурирати интервал као што је раније описано.

Напоменимо да се неки од ових корака требају променити или прескочити у случају да је поље на левој или десној ивици интервала . Са тренутном имплементацијом операције ,  која ради у сложености , неопходно је сваки пут поново сумирати све неопходне елементе низа када имамо команду ; ово је сложености , тако да је приступ и даље неефикасан и доноси ~60 бодова.

Да бисмо ефикасније одређивали суме првих елемената низа , неопходно је користити структуру која се зове **Фенвиково стабло** *(Fenwick tree / Binary Indexed Tree / BIT)* које нам омогућава да рачунамо ове суме у временској сложености по цену тога да и операција мора бити сложености . Ово нам даје укупну временску сложеност , која је довољна за добијање 100 бодова. Више о структури Фенвиковог стабла можете наћи у **додатку А**.