|  |  |  |
| --- | --- | --- |
| **Тагове** | **На пълното решение** | **На подзадачите** |
| Делимост | Пълно изчерпване  Алгоритъм на Евклид |

**Анализ**

**Решение за 25 точки**

За всяка възможна пермутация намираме най-голямото , за което може песните да се разпределят в плейлисти. Как по-точно? Всъщност, ние бихме могли да разпределим песните от жанрове и в плейлисти с големина , само ако се дели на . Ние искаме да намерим НОД-а на ,,, …, , което може да стане с алгоритъма на Евклид. Забележете, ако , не трябва да гледаме сбора на при намирането на , а само .

Постигната сложност:

Имплементация: brute.cpp

**Решение за 75 точки**

Нека е максималното число в редицата, а – минималното число. , защото иначе няма как да направим плейлист, който включа жанра, който е с най-малко песни. Нека разгледаме всички числа от до и да проверим за всяко едно дали е възможна големина на плейлисти.

Нека сме да искаме да проверим, дали е възможна големина. Нека за случайни и , да се дели на . Тогава също се дели на (с означавам остатъка на при деление на ). Нека . Нека също е броят елементи в , равни на . Ако успеем да намерим подходящо групиране за , то ще важи също и за . Ако няма възможно групиране за , то няма и за . За всички не пречи . За някое , , то задължително . Аналогично, за някое , , то задължително . **За** **всяко го групираме с някое друго , за което .** Забележете, ако е четно и , то и . **Тоест, ако е четно, задължително трябва да се дели на .** За всички останали стойности в , задължително трябва . Когато тези условия се спазват, е възможна големина на плейлистите.

Постигната сложност:

Имплементация: slow.cpp

**Решение за 100 точки**

Нека съкратим бройката на възможните големини на плейлисти. За да e валидно някое , то трябва задължително да дели (всяка песен участва в плейлист и всеки плейлист е с големина ). Разглеждаме всички делители на сбора на числата на редицата и проверяваме всички, които са .

Постигната сложност:

Имплементация: author.cpp и alternative.cpp

Сложността на решението е такава, защото максималния брой делители на число е приблизително . Решението се държи и по-бързо, защото не всички делители на са . Имплементация може да намерите в файла author.cpp.

Едно допълнение – докато правих тестовете, забелязах, че при решението, което разглежда делителите на сумата, няма нужда да се проверява дали се дели на . Ако дели сумата и за всяко , то задължително е четно.

*Автор: Борис Михов*