

XXXV НАЦИОНАЛНА ОЛИМПИАДА ПО ИНФОРМАТИКА

Национален кръг

Стара Загора, 15 – 17 март 2019 г.

Група АВ, 9 – 12 клас, Ден 2

Задача АВ4. СУМИ ПО ДВОЙКИ

Ани и Боби играят следната игра. Ани винаги е първа. Тя си измисля N цели положителни числа $a_1, a_2, \dots, a_N$ и съставя сумите на всеки две от тях, като така получава $N*(N-1)/2$ суми. Тях подрежда във възходящ ред и ги дава на Боби. Той трябва да познае оригиналната редица от числа. Ако ги познае, той печели, иначе печели Ани.

Това дали Боби е познал се определя от Ани. Тоест Боби казва на Ани - "Смятам, че това е била оригиналната редица". Ани казва "Позна" или "Не, оригиналната редица е следната...". По този начин Боби може реално да е познал, но Ани да каже, че е имала предвид друга редица със същите суми по двойки. Например:

Ани: Намисля си редицата "1 4" и казва на Боби "5" (редицата от всички суми по двойки).

Боби: Смятам, че оригиналната редица е "1 4".

Ани: Не, оригиналната редица е "2 3".

Печели Ани.

Ани: Намисля си редицата "1 2 3" и казва на Боби сумите "3 4 5".

Боби: Смятам, че оригиналната редица е "1 2 3".

Ани: Позна.

Печели Боби.

Помогнете на Боби като напишете програма **pairwise**, която по дадена редица със суми по двойки, намира броя на възможните първоначални редици, както и лексикографски най-малката от тях.

Вход:

От първия ред на стандартния вход ред се въвежда цялото положително число N . От следващия ред се въвеждат $N*(N-1)/2$ числа – сумите по всички двойки от числата, които Ани е измислила, подредени във възходящ ред.

Изход:

На първия ред на стандартния изход изведете едно число - броя на различните редици, които Ани може да си е намислила. На втория ред изведете N числа, подредени в нарастващ ред - лексикографски най-малката редица от числа, които образуват суми по двойки, равни на въведените.

XXXV НАЦИОНАЛНА ОЛИМПИАДА ПО ИНФОРМАТИКА

Национален кръг

Стара Загора, 15 – 17 март 2019 г.

Група АВ, 9 – 12 клас, Ден 2

Ограничения

$3 \leq N \leq 7$	$0 < a_i \leq 10^9$	20% от тестовете
$3 \leq N \leq 180$	$0 < a_i \leq 100$	25% от тестовете
$3 \leq N \leq 180$	$0 < a_i \leq 10^9$	45% от тестовете
$3 \leq N \leq 500$	$0 < a_i \leq 10^9$	10% от тестовете

Оценяване

Всеки тест се оценява отделно.

Пример

Вход	Изход
3	1
3 4 5	1 2 3

XXXV НАЦИОНАЛНА ОЛИМПИАДА ПО ИНФОРМАТИКА

Национален кръг

Стара Загора, 15 – 17 март 2019 г.

Група АВ, 9 – 12 клас, Ден 2

Задача АВ5. КВАДРАТ

Дадена е правоъгълна таблица с N реда и M колони. В началото всички полета от таблицата са бели. Следват Q заявки, на всяка от които едно или няколко последователни полета, които са на един ред или стълб, се оцветяват в черно. Ако едно поле поне веднъж се оцвети в черно, то накрая ще е оцветено в черно.

Да се намери страната на най-големия квадрат, който се състои изцяло от бели полета след изпълнението НА ВСИЧКИ заявки за оцветяване. Горният ляв ъгъл на таблицата е с координати $(1,1)$.

Вход

От първия ред на стандартния вход се въвеждат две цели положителни числа N и M – броя на редовете и колоните на таблицата.

От втория ред се въвежда положително цяло число Q – броя на заявките.

Следват Q реда с по четири цели числа X_{i1} , Y_{i1} , X_{i2} , Y_{i2} . Всички полета (X, Y) , за които $X_{i1} \leq X \leq X_{i2}$ и $Y_{i1} \leq Y \leq Y_{i2}$ се оцветяват в черно.

Изход

На стандартния изход изведете единствено число – дължината на страната на най-големия квадрат съставен от изцяло бели полета.

Ограничения

- ♣ $1 \leq N, M \leq 10^9$
- ♣ $1 \leq Q \leq 50\,000$
- ♣ $1 \leq X_{i1} \leq X_{i2} \leq N$, за всяко i : $1 \leq i \leq Q$
- ♣ $1 \leq Y_{i1} \leq Y_{i2} \leq M$, за всяко i : $1 \leq i \leq Q$
- ♣ $X_{i1}=X_{i2}$ или $Y_{i1}=Y_{i2}$, за всяко i : $1 \leq i \leq Q$

Подзадачи

Подзадача	Точки	N	M	Q	Други
1	10	≤ 100	≤ 100	$\leq 1\,000$	
2	10	≤ 1000	≤ 1000	$\leq 50\,000$	
3	10	≤ 100	$\leq 10^5$	$\leq 50\,000$	
4	10	≤ 100	$\leq 10^6$	$\leq 50\,000$	
5	20	$\leq 10^5$	$\leq 10^5$	$\leq 30\,000$	$X_{i1}=X_{i2}$, $Y_{i1}=Y_{i2}$, за всяко i : $1 \leq i \leq Q$
6	25	$\leq 10^5$	$\leq 10^5$	$\leq 30\,000$	
7	15	$\leq 10^9$	$\leq 10^9$	$\leq 50\,000$	

За да се получат точките за всяка от подзадачите, трябва програмата ви да даде верни отговори на всички тестове за подзадачата.

XXXV НАЦИОНАЛНА ОЛИМПИАДА ПО ИНФОРМАТИКА

Национален кръг

Стара Загора, 15 – 17 март 2019 г.

Група АВ, 9 – 12 клас, Ден 2

Пример

Вход	Изход	Обяснение на примера				
6 5 4 1 2 1 5 1 4 5 4 5 2 5 2 1 2 1 5	3		X	X	X	X
					X	
					X	
					X	
		X			X	

XXXV НАЦИОНАЛНА ОЛИМПИАДА ПО ИНФОРМАТИКА

Национален кръг

Стара Загора, 15 – 17 март 2019 г.

Група АВ, 9 – 12 клас, Ден 2

Задача АВ6. СЪОБЩЕНИЕ

Марсоходът Опъртюнители иска да изпрати съобщение до Земята. Проблемът е, че уредите му са повредени и части от съобщението няма успешно да се изпратят. По-конкретно, информацията, която той желае да достигне Земята, се състои от N бита. Марсоходът може да изпрати съобщение с избрана от него дължина M бита. Още преди изпращането на съобщението е известно, че точно D бита от него ще бъдат изгубени. Марсоходът иска да изпрати такова съобщение, че на Земята да могат да възстановят максимално точно ползватата информация от N бита, която той желае да предаде.

Един от ръководителите на мисията Ви е помолил за помощ – да измислите и имплементирате протокола за комуникация. Трябва да напишете функция *transmit*, която Опъртюнители ще използва. Тя получава низ от N бита и стойността на D и трябва да генерира друг низ с дължина M – това са данните, които марсоходът ще опита да излъчи. Също така трябва да напишете и отделна функция *receive*, която NASA ще използва на Земята. Тя получава низ от $M - D$ бита – съобщението генерирано от *transmit* след загубата на някои от битовете, както и стойностите на N и на D . Тя трябва да реконструира оригиналните данни от N бита максимално точно.

Целта е двойката функции да максимизират отношението на успешно реконструирана смислена информация (а не просто шум) към общото количество успешно изпратена информация. Това ще се оценява по следната формула (*correct* е броят правилно познати битове):

$$\frac{2 \times \text{correct} - N}{\max(M - D, N)}$$

Детайли по имплементацията

Функция *transmit* трябва да има следния прототип:

```
std::vector<bool> transmit(const std::vector<bool>& data, int d);
```

Тя ще бъде извикана точно веднъж и ще получи като аргументи данните за изпращане и бройката битове, които ще бъдат изтрити, като трябва да върне съобщението за изпращане.

Функцията *receive* трябва да има следния прототип:

```
std::vector<bool> receive(const std::vector<bool>& message, int n, int d);
```

Тя също ще бъде извикана точно веднъж и ще получи като аргументи полученото на Земята съобщение и стойностите на N и D , като трябва да върне вектор с дължина точно N – реконструираното с някаква точност съобщение.

Двете функции задължително трябва да бъдат имплементирани в отделни файлове **transmit.cpp** и **receive.cpp**, които трябва да изпратите към системата (за тази задача системата предоставя възможност да се изпращат два файла). В тях може да имате каквито искате помощни функции, структури, променливи и т.н. Силно препоръчително е всички такива (т.е. всичко без самите функции *transmit* и *receive*) да бъдат декларирани като *static* с цел да се предотвратят конфликти в имената между двата ваши файла). Те **не трябва** да съдържат функция *main* и **трябва** да включват хедър файла **transmission.h** чрез указание към препроцесора `#include "transmission.h"` в началото.

XXXV НАЦИОНАЛНА ОЛИМПИАДА ПО ИНФОРМАТИКА

Национален кръг

Стара Загора, 15 – 17 март 2019 г.

Група АВ, 9 – 12 клас, Ден 2

ВАЖНО: Двете функции ще бъдат изпълнявани в различни процеси на системата и НЕ могат да обменят информация пряко помежду си.

Ограничения

$N = 10\,000$ във всички тестове

$1 \leq D \leq N/10$

$M \leq 100\,000$

Оценяване

Всеки тест се оценява отделно. За даден тест Вашето решение ще получи точки, различни от 0, ако двете функции успешно приключат изпълнението си, функцията *transmit* върне вектор с дължина не по-голяма от 100 000 и функцията *receive* върне вектор с вярна дължина и е познала над $\frac{1}{2}$ от битовете. Точките, които ще получите на теста са равни на максималният брой точки за теста, умножени по $\min(\text{yourScore}/\text{authorScore}, 1)$, където *yourScore* е резултатът Ви, пресметнат по формулата, дадена по-горе, а *authorScore* е резултатът на авторовото решение, пресметнат по същата формула. Части от тестовете имат допълнителни ограничения.

Тестове

Низовете от данни за изпращане са произволно генерирани за всички тестове, бит по бит, където 1 и 0 имат равен шанс.

Кои битове ще бъдат изтрети не зависи по смислен начин от съдържанието на съобщението, което изпращате, а само от размера му, но може да се мени при различни съобщения. Позициите на триене са произволно генерирани на по групи от съседни битове с дължина K .

В 15% от тестовете: $D = 1$ и $K = 1$ **минимален резултат:** 0.971

В 25% от тестовете: $D \leq 1.1 \times \sqrt{N}$ и $K = 1$ **минимален резултат:** 0.771

В 25% от тестовете: $K = 1$ **минимален резултат:** 0.473

В 35% от тестовете: без доп. ограничения **минимален резултат:** 0.432

За всяка група Ви е даден минималният резултат, който авторовото решение е получило на тест от нея, с точност до третия символ след десетичната запетая.

Локално тестване

Предоставени Ви са файловете **transmission.h** и **Lgrader.cpp**, които можете да компилирате заедно с Вашите програми, за да ги тествате, както и две примерни глупави имплементации на функциите. За да компилирате файловете заедно използвайте следната команда в конзолата/терминала (командата трябва да се изпълни в папката, в която се намират файловете):

```
g++ -O2 -std=c++11 -o transmission.exe receive.cpp transmit.cpp Lgrader.cpp
```

При стартиране програмата ще Ви пита за N , D и K и от там всичко друго ще се генерира произволно. Ако искате да я конфигурирате по друг начин, може да правите каквито си промени искате по предоставените Ви файлове.

XXXV НАЦИОНАЛНА ОЛИМПИАДА ПО ИНФОРМАТИКА

Национален кръг

Стара Загора, 15 – 17 март 2019 г.

Група АВ, 9 – 12 клас, Ден 2

Примерна комуникация

№	Викания на функции от журито	Техният return
1.	transmit({0,0,1,0,1,0},2)	{0,0,1,0,1,0}
2.	receive({0,1,1,0},6,2)	{0,0,0,0,0,0}

Обяснение на примерната комуникация

Тази примерна комуникация се е извършила с двете примерни наивни имплементации на функциите, които са Ви предоставени.

Данните за изпращане са 001010 и ще бъдат изтрети два бита. Функцията transmit не прави никакви модификации по съобщението и се опитва да изпрати просто това. Не са се изпратили успешно вторият и четвъртият бит и receive получава 0110. Тази наивна имплементация обаче изобщо не опитва да реконструира данните, ами просто връща низ от нули.

Резултатът, който двойката функции получават на теста, е $(2 \times 4 - 6)/6 = 1/3$. Точките, които ще се получат, зависят от резултата на решението на автора.