ВТОРО КОНТРОЛНО СЪСТЕЗАНИЕ
НА РАЗШИРЕНИЯ НАЦИОНАЛЕН ОТБОР
ВЕЛИКО ТЪРНОВО, 08 МАЙ, 2016 Г.
ГРУПА А

Задача АК3. ПАРК

Автор: Антон Анастасов

Натали обича високите скорости и е решила да прекара следобеда си в най-големия и модерен лунапарк в България – този във Велико Търново, който тъкмо днес отваря за първи път вратите си за посетители. И понеже "Виенското колело", "Гондолата" и "Блъскащите колички" са доста скучни за нейния авантюристичен характер, тя е твърдо решена да не губи време, а да се забавлява на "Влакчето на Адреналина".

"Влакчето на Адреналина" е много по-модерно от всички влакчета на ужасите, за които някога сте чували през живота си. То се състои от N станции, номерирани с числата от 1 до N, като от всяка станция излиза *точно една* еднопосочна линия към някоя от другите $N-1$ станции. Станциите са три вида:

- Червени : червена е такава станция, тръгвайки от която влакчето отново се връща в нея, след като премине през някакъв брой (по-голям от 0) други станции, следвайки еднопосочните линии;
- Жълти: жълта е такава станция, която не е червена, но еднопосочната линия, тръгваща от нея, води в червена станция;
- Зелени: всички останали станции са зелени.

Едно пътуване с това модерно влакче винаги трябва да започва от зелена или жълта станция и да завършва в червена, като не трябва да преминава повече от един път през една и съща станция (дори и като крайна спирка).

Преминаването през всяка линия между две станции носи определено количество адреналин. Това количество може да бъде както положително, така и отрицателно.

Абсолютна новост при това влакче е, че, от време на време, персоналът променя структурата на мрежата от линии, по които то се движи (така то става много по-атрактивно). Промяната, която може да се прави, е много специфична: *вземат се две жълти станции и се разменят краищата (червените станции) на линиите, които излизат от тях. При тази размяна се запазват количествата адреналин, които носят двете линии, излизащи от двете жълти станции*.

И така, Натали е при „Влакчето на Адреналина“ и иска да опита различни маршрути. Тя си избира определена червена станция с номер F, до която иска да стигне, и веднага пред нея възниква въпросът: на коя жълта или зелена станция да се качи, така че с едно пътуване да стигне до станция с номер F, като общото количество адреналин, което ще получи при това пътуване да бъде поне K (разбира се, такава станция може и да не съществува).

Задача

Напишете функция *solve()*, която ще се компилира заедно с програмата на журито и трябва да изпълни Q заявки, които са от два типа:

- тип 1: въпрос от Натали за определяне на начална станция за поредното пътуване;
- тип 2: заявка от персонала за промяна структурата на мрежата.

ВТОРО КОНТРОЛНО СЪСТЕЗАНИЕ
НА РАЗШИРЕНИЯ НАЦИОНАЛЕН ОТБОР
ВЕЛИКО ТЪРНОВО, 08 МАЙ, 2016 Г.
ГРУПА А

Детайли по реализацията

Функцията *solve()* трябва да има следния формат:

*void solve(int N, int Q, int *next, int *weight)*

Тази функция се вика веднъж от програмата на журито и нейните параметри са:
N – брой на станциите;
Q – брой на заявките;
next[] – масив, задаващ структурата на мрежата от линии на влакчето в момента, в който Натали се появява в атракциона – *next[i](0≤i<N)* съдържа номера на станцията, до която води еднопосочната линия, излизаща от станция с номер *i+1*. *Внимание: не бъркайте индексацията на масива next (която е от 0 до N-1) с номерацията на станциите (която е от 1 до N).*
weight[] – масив, задаващ количествата адреналин по различните линии – *weight[i]* съдържа количеството адреналин, което се получава при преминаване по линията, която излиза от станция с номер *i+1*.

За комуникация с програмата на журито ви се предоставят следните функции:

*int getQuery(long long *arg0, long long *arg1)*

Тази функция трябва да се вика *Q* пъти, като при всяко извикване се получава заявка, която трябва да бъде изпълнена:

- ако функцията *getQuery()* върне 1, това е въпрос от Натали и в този случай *arg0* е номерът на червената станция, до която момичето иска да стигне, а *arg1* – минималното количество адреналин, което иска да получи. За да отговорите на заявка от този тип, ви се предоставя функцията
  void answerQuery(int answerType, int answer).

  Ако не съществува начална станция, удовлетворяваща желанието на Натали, трябва да отговорите с *answerQuery(0,-1)*, а ако съществува, с *answerQuery(1, start)*, където *1≤start≤N* е номерът на една от възможните начални станции.

- ако функцията *getQuery()* върне 2, това е заявка от персонала за промяна структурата на мрежата и в този случай *arg0* и *arg1* съдържат номерата на жълтите станции, за които ще бъдат разменени „червените“ краища на излизащите от тях линии. В този случай нищо не трябва да отговаряте, а просто да извършите поисканата промяна.

Вие трябва да предадете към системата файл **park.cpp,** който съдържа функция *solve()*. Той може да съдържа и друг код, необходим за работата на функция *solve()*, но не трябва да съдържа *main()*.

В началото си Вашият файл трябва да съдържа #include "park.h".

Ограничения

abs(weight[i]) <= 1'000'000'000

Останалите ограничения са дадени в подзадачите.

ВТОРО КОНТРОЛНО СЪСТЕЗАНИЕ
НА РАЗШИРЕНИЯ НАЦИОНАЛЕН ОТБОР
ВЕЛИКО ТЪРНОВО, 08 МАЙ, 2016 Г.
ГРУПА А

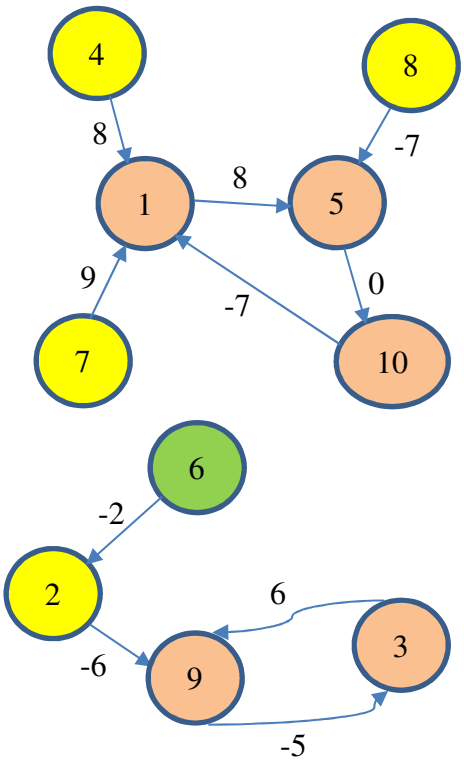
Пример

Стъпка №	**Действие**	**Обяснение**
1	Програмата на журито вика вашата функция *solve()* с аргументи: *N*=10, *Q*=7 и масиви: *next*: 5,9,9,1,10,2,1,5,3,1 *weight*: 8,-6,6,8,0,-2,9,-7,-5,-7	Началната мрежа от линии изглежда като на фигура 1 със съответните количества адреналин. Станции 1, 3, 5, 9, 10 са червени, станции 2, 4, 7, 8 са жълти, а станция 6 е единствената зелена.
2	Викате функция *getQuery()* и тя връща 1 и *arg0*=5, *arg1*=20. Вашата програма отговаря с извикване на *answerQuery(0,-1)*	Заявката е от тип 1 – Натали пита от коя жълта или зелена станция да тръгне, така че като стигне до червена станция 5 да събере адреналин поне 20. Вие отговаряте, че няма такава станция.
3	Викате функция *getQuery()* и тя връща 1 и *arg0*=5, *arg1*=17. Вашата програма отговаря с извикване на *answerQuery(1,7)*	Заявката е от тип 1 – желаната крайна станция е 5 и минимален адреналин 17. Вие отговаряте, че, ако тръгне от станция 7, работата ще стане (7->1->5).
4	Викате функция *getQuery()* и тя връща 1 и *arg0*=3, *arg1*=-20. Вашата програма отговаря с извикване на *answerQuery(1,6)*	Заявката е от тип 1 – желаната крайна станция е 3 и минимален адреналин -20. Вие отговаряте, че, ако тръгне от станция 6, работата ще стане (6->2->9->3).
5	Викате функция *getQuery()* и тя връща 1 и *arg0*=9, *arg1*=2. Вашата програма отговаря с извикване на *answerQuery(0,-1)*	Заявката е от тип 1 – желаната крайна станция е 9 и минимален адреналин 2. Вие отговаряте, че няма такава станция.

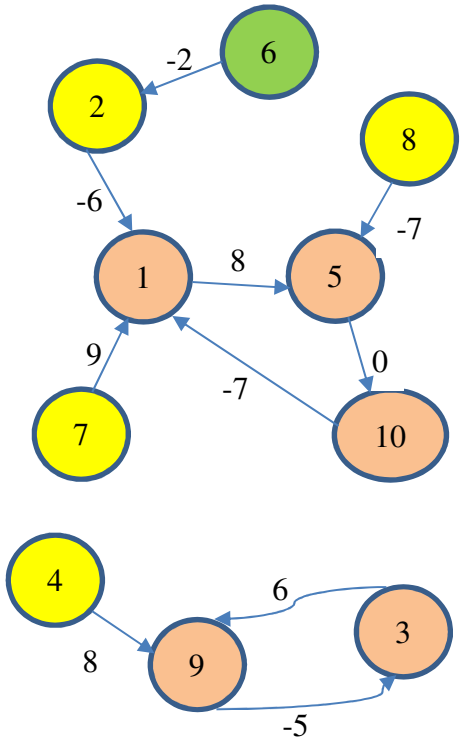
ВТОРО КОНТРОЛНО СЪСТЕЗАНИЕ
НА РАЗШИРЕНИЯ НАЦИОНАЛЕН ОТБОР
ВЕЛИКО ТЪРНОВО, 08 МАЙ, 2016 Г.
ГРУПА А

6	Викате функция *getQuery()* и тя връща 2 и *arg0*=2, *arg1*=4.	Това е заявка от персонала за промяна на линиите, излизащи от жълти станции 2 и 4. Нищо не отговаряте, а променяте мрежата – тя придобива вида, показан на фигура 2.
7	Викате функция *getQuery()* и тя връща 1 и *arg0*=9, *arg1*=2. Вашата програма отговаря с извикване на *answerQuery(1,4)*	Заявката е от тип 1 – желаната крайна станция е 9 и минимален адреналин 2. Заявката е същата, като предния въпрос на Натали, но структурата на мрежата е променена и тръгването от станция 4 осигурява искания адреналин (4->9).
8	Викате функция *getQuery()* и тя връща 1 и *arg0*=1, *arg1*=-20. Вашата програма отговаря с извикване на *answerQuery(1,6)*	Заявката е от тип 1 – желаната крайна станция е 1 и минимален адреналин -20. Вие отговаряте, че, ако тръгне от станция 6, работата ще стане (6->2->1).

Фигура 1



Фигура 2



ВТОРО КОНТРОЛНО СЪСТЕЗАНИЕ
НА РАЗШИРЕНИЯ НАЦИОНАЛЕН ОТБОР
ВЕЛИКО ТЪРНОВО, 08 МАЙ, 2016 Г.
ГРУПА А

Подзадачи

Подзадача	Точки	N	Q	Бележки
1	2	≤ 1 000	≤ 100	Без заявки от тип 2
2	3	≤ 1 000	≤ 50 000	Без заявки от тип 2
3	7	≤ 50 000	≤ 50 000	Без заявки от тип 2
4	17	≤ 500 000	≤ 500 000	Без заявки от тип 2
5	3	≤ 1 000	≤ 100	
6	5	≤ 1 000	≤ 10 000	
7	21	≤ 50 000	≤ 50 000	
8	42	≤ 500 000	≤ 500 000	

Локално тестване

За да тествате решението си на локалния компютър, Ви се предоставят файлове **Lgrader.cpp** и **park.h**. Компилирайте ги заедно с вашия файл **park.cpp** и ще получите изпълнима програма, която може да обработва списък от заявки.

Входът за локалния грейдър изглежда по следния начин:

От първия ред се въвеждат *N* и *Q*.

От втория ред се въвежда масивът *next*.

От третия ред се въвежда масивът *weight*.

От всеки от следващите *Q* реда се въвежда по една заявка, описана с три цели числа queryType, arg0, arg1: queryType е 1 или 2, където 1 е заявка от страна на Натали, а 2 - от страна на персонала; arg0 и arg1 следват формата описана по-горе.

Локалният грейдър извежда на стандартния изход следните данни:

На първи ред – брой на заявките *Q*.

За всяка заявка от тип 1 извежда един ред, на който пише 0, ако няма търсения път, или 1 *x*, където *x* е станцията, от която трябва да се тръгне (това са отговорите, които Вашата функция *solve()* е намерила).

За всяка заявка от тип 2 извежда просто 2.