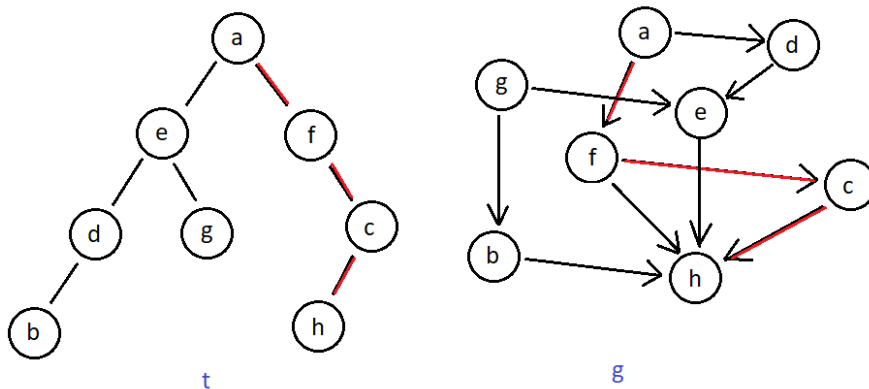


Задача 1. Нека имаме двоично дърво от цели числа представено или като обект от клас (например `BinTree`), или с указател към корена на дървото. Да се реализира външна функция, която приема едно такова дърво и пресмята сумата на елементите от четните нива, които имат *само* ляв наследник.

Забележка: приемаме, че коренът е на ниво 0 и това е четно ниво.



Задача 2. Нека имаме ориентиран граф от символи g и двоично дърво от символи t . Множеството от елементите на двете структури е едно и също, като в рамките на всяка една от структурите символите са **уникални**.

Напишете функция, която проверява дали съществува път от корена на дървото до листо, който да отговаря на път в графа и ако има такъв го извежда (**Заб.:** достатъчно е да изведете само един път, който отговаря на условието).

Пример:

В случая пътят $a - f - c - h$ отговаря на условието.

Задача 3. Създайте (чрез включване) двоично наредено дърво от интервали от числа ($[a, b]$, $a \leq b$).

Наредбата в дървото да бъде по средната стойност на интервала.

Намерете максималния интервал от дървото (спрямо описаната наредба в дървото), чието сечение с всеки един от другите интервали интервали на двоичното наредено дърво не е празно или върнете $[0,0]$, ако такъв не съществува. Използвайте наредеността на дървото, където е удачно.

Заб.: можете да използвате реализация на двоично наредено дърво или ако предпочитате да използвате указател към корен на дървото.

=====

1. Да се реализира абстрактен клас `TrainStation`, който описва ЖП гара и поддържа следните операции:

- `bool link ([попълнете правилния тип] station)` – установява връзка между дадената гара и друга гара `station`. Създадената връзка трябва да е двупосочна. Операцията не е възможна, ако поне една от двете гари не може да бъде свързана с повече гари. В този случай методът връща `false`.
- `[попълнете правилния тип] getLinkedStation (int i)` – връща i -тата поред гара (при избрана от вас подредба), към която дадената гара е свързана, и `nullptr`, ако дадената гара е свързана с по-малък брой гари.

2. Да се реализират наследници `Terminal` и `Hub` на `TrainStation`. За `Terminal` е характерно, че гарата може да е свързана най-много с още една гара. За `Hub` е характерно, че гарата може да е свързана с максимум 50 други гари.

3. Да се реализира функция `bool connected ([попълнете правилния тип] s1, [попълнете правилния тип] s2)`, която проверява дали има връзка (пряка или косвена) между двете гари `s1` и `s2`.

4. Да се реализира подходяща функция `serialize`, сериализираща цяла мрежа от гари, т.е. функция, която да съхрани информацията за гарите и техните връзки във файл по избор. Не се изисква реализация на функция за четене на записана мрежа от гари, но избраният формат за сериализация трябва да позволява така възможност (т.е. представянето да е пълно и еднозначно).