



Polsemestrálny test praktická časť



Zdrojový kód pre prvé dve úlohy nájdete na stránke, resp. v moodli. Evaluátor vyžaduje tieto názvy tried a tiež názvy metód, ktoré sú zadané. Všetky ostatné názvy premenných, vnorených tried a ostatných pomocných metód si môžete upraviť.



1. Reklamné prestávky (5b) Televízne stanice majú prísne pravidlá - ak sled niekoľkých relácií bol bez prerušenia, diváci musia byť „odmenení“ reklamou. V našom televíznom programe sú jednotlivé bloky zadané v minútach ako spájaný zoznam. Vašou úlohou je zabezpečiť, že vždy, keď by pridanie ďalšieho bloku prekročilo stanovenú hranicu, vloží sa reklamná prestávka ešte pred týmto blokom.

Do triedy SpajanyZoznam pridajte metódu `commercialTime`, ktorá dostane ako parameter dĺžku reklamy v minútach a hranicu, podľa ktorej rozhodne, či po slede niekoľkých programoch má nasledovať reklama, alebo nie. Môžeme uvažovať, že žiaden z programov nemá dĺžku väčšiu, ako zadaná hranica.

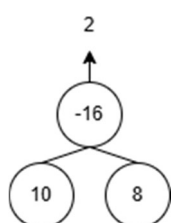
public void commercialTime(int length, int threshold)

Metóda nech pracuje v lineárnom čase vzhľadom k dĺžke zoznamu, t.j. $O(n)$, kde n je dĺžka zoznamu. Za opakovaný prechod spájaným zoznamom je bodový zisk znížený o 1 bod.

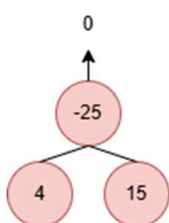
Uvažujeme program [30, 20, 50, 10, 15, 40]: Metóda zmení obsah tohto zoznamu takto:

- `commercialTime(5, 60)` → [30, 20, 5, 50, 10, 5, 15, 40]
(pretože $30+20+50 > 60$, $50+10+15 > 60$)
- `commercialTime(3, 50)` → [30, 20, 3, 50, 3, 10, 15, 3, 40]
(pretože $30+20+50 > 50$, $50+10 > 50$, $10+15+40 > 50$)
- `commercialTime(10, 100)` → [30, 20, 50, 10, 10, 15, 40]
(pretože $30+20+50+10 > 100$)

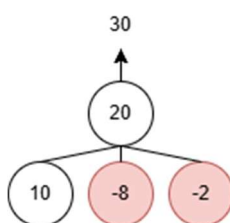
2. Konsolidácia (5b) V období konsolidácie musí štát aj spoločnosti likvidovať subjekty, ktoré neprinášajú profit. Tieto oddelenia budeme reprezentovať stromovou štruktúrou (nie binárnou), v ktorej jednotlivé oddelenia sú znázornené ako uzol s hodnotou predstavujúcou jeho bilanciu. Počas plánovania procesu konsolidácie chceme spočítať, aký bude finančný stav spoločnosti po tom, ako sa zlikvidujú tie časti, ktoré sú stratové. Ak sa spoločnosť rozhodne zrušiť oddelenie, tak zaniknú aj všetky jej podradené subjekty.



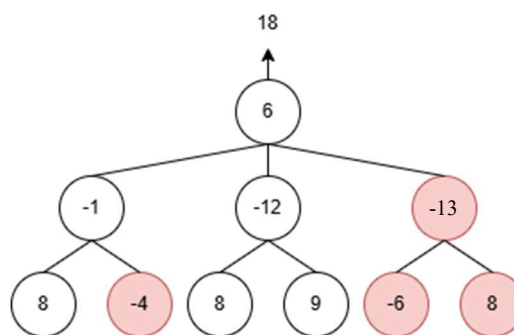
Priklad 1



Priklad 2



Priklad 3



Priklad 4

Pri konsolidácii sa nehľadí na samostatné uzly, ale na celé štruktúry: nevadí ak jedno oddelenie je stratové ak ho “zachráni” jemu podriadené subjekty. Modelové príklady znázorňujú aký zisk vytvárajú štyri rôzne firmy po vykonaní konsolidácie.

Uvažujme triedu Oddelenie, ktorá uchováva hodnotu samotného oddelenia a zoznam podriadených oddelení. Implementujte metódu, ktorá vypočíta bilanciu spoločností po odrezaní stratových oddelení. Samotnú štruktúru nie je potrebné modifikovať.

```
public int totalProfit()
```



Ilustračný obrázok

3. **PAZwave** (5b) IP adresa sa používa na identifikáciu zariadení v sieti. Jej verzia 4 sa zvyčajne zapisuje ako štvorica čísel z intervalu 0-255, oddelených bodkami. **PAZwave v.X** je špeciálny protokol na identifikáciu zariadení v sieti **PAZ-LAN**, ktorý pozostáva z X-tice čísel takisto z intervalu 0-255, oddelených bodkami. Počas spracovania sieťových logov nastal problém - PAZwave adresy boli omylom uložené ako jedno dlhé číslo, bez bodiek. Zostala však informácia o tom, o akú verziu protokolu šlo (teda hodnota X).

Vašou úlohou je zistiť, koľko rôznych platných PAZwave adries verzie X môže vzniknúť rozdelením tohto čísla na X častí zodpovedajúcich číslam v rozsahu 0-255. Pre hodnotu parametra x platí $0 < x < \text{počet cifier zadaného čísla}$.

V triede PAZwave vytvorte metódu `validAddressVersionX`, ktorá vráti počet rôznych platných PAZwave adries, ktoré mohli vzniknúť rozdelením zadaného čísla reprezentovaného ako objekt triedy String na chunky:

```
public int validAddressVersionX(String number, int x)
```

Príklady:

- `validAddressVersionX("255255255255", 4) -> 1 (255.255.255.255)`
- `validAddressVersionX("11111", 3) -> 6 (1.1.111, 1.11.11, 1.111.1, 11.1.11, 11.11.1, 111.1.1)`
- `validAddressVersionX("33333", 3) -> 3 (3.33.33, 33.3.33, 33.33.3)`
- `validAddressVersionX("11311", 3) -> 5 (1.13.11, 1.131.1, 11.3.11, 11.31.1, 113.1.1)`