Úloha 1

(10 bodů) Na vstupu je zadána posloupnost **kladných** celých čísel délky n. Čísla si zapíšeme v zadaném pořadí na kružnici ve směru hodinových ručiček, takže za číslem a_{n-1} následuje číslo a_0 (pozice čísel ve vstupní posloupnosti číslujeme od 0). Určete, jak lze kružnici s čísly rozdělit na dva souvislé úseky se **stejným součtem čísel**. Každé z čísel bude tedy náležet do právě jednoho úseku. Výstupem jsou index prvního a index posledního prvku v jednom a ve druhém úseku (opět v pořadí směru hodinových ručiček). Má-li úloha více řešení, stačí nalézt jedno libovolné z nich.

Navrhněte postup, jak správně vyřešit úlohu **s co nejlepší časovou složitostí** vzhledem k délce vstupní posloupnosti. Plný počet bodů bude udělen pouze za asymptoticky optimální řešení.

(a) **Popište algoritmus**: programový kód v Pythonu je vítán, ale není povinný, slovní vysvětlení zvoleného postupu řešení naopak povinné je. Nepoužívejte žádné netriviální datové struktury (typu zásobník, fronta, halda, slovník), jejichž algoritmus sami nepopíšete a neodvodíte jeho časovou složitost.

(b) **Zdůvodněte správnost** algoritmu.

(c) Odvoďte **časovou a prostorovou složitost** (v nejhorším případě). Pracuje vaše řešení v asymptoticky optimálním čase a prostoru?

Úloha 2

(10 bodů) Je zadán binární strom o n vrcholech, v nichž jsou uložena navzájem různá celá čísla od 1 do n. Navrhněte efektivní algoritmus, který určí čísla **všech listů s minimální vzdáleností od kořene stromu**.

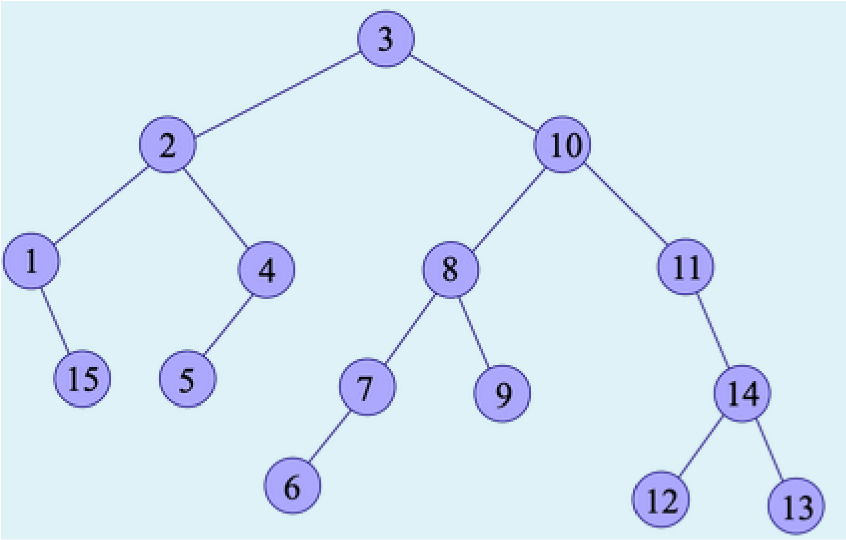
(a) Svoje řešení zapište jako **funkci v Pythonu**, využijte definici třídy pro vrchol binárního stromu níže a váš kód prosím opatřete **komentáři**,

(b) zdůvodněte **správnost**,

(c) odvoďte **časovou složitost**.

```
class VrcholBinStromu:
    """třída pro reprezentaci vrcholu binárního stromu"""
    def __init__(self, x = None, levy = None, pravy = None)
        self.info = x      # očíslování vrcholů
        self.levy = levy   # levé dítě
        self.pravy = pravy # pravé dítě
```

Příklad

Vstup: 

Výstup: 15 5 9

Úloha 3

Odpovězte na otázky, své odpovědi vždy **zdůvodněte**.

(a) **(3 body)** Mějme dánu trojici funkcí $f_1, f_2, g : \mathbb{N} \to \mathbb{R}^+$ splňující $f_1 = O(g)$, $f_2 = O(g)$. Dokažte nebo vyvraťte každé z následujících tvrzení:

- $f_1 \cdot f_2 = O(g)$
- $f_1 \cdot f_2 = O(g^2)$
- $\frac{f_1}{f_2} = O(g^2)$

(b) **(3 body)** V binárním vyhledávacím stromu chceme implementovat operaci **odebrání prvku** s danou hodnotou klíče x následovně: Vyhledáme ve stromu vrchol v s hodnotou x (předpokládejme, že to umíme a že se x skutečně ve stromu nachází) a pak opakovaně provádíme následující kroky výpočtu:

- je-li vrchol v listem, odebereme ho ze stromu a tím celé operace **končí**
- není-li vrchol v listem, označíme symbolem s jeho jednoho potomka (levého, má-li oba)
- hodnotu z vrcholu s zkopírujeme do vrcholu v
- symbolem v nyní označíme vrchol s

Rozhodněte, zda výše popsaný postup korektně implementuje požadovanou operaci.

(c) **(3 body)** Uvažte **strom hry** dvou hráčů, bílého a černého, jehož listy byly ohodnoceny hodnotami 1 (pokud vyhrál bílý), -1 (vyhrál černý) a 0 (remíza). Předpokládejme, že první je na tahu bílý, a že po ohodnocení zbylých vrcholů stromu **minimaxovým algoritmem** byl nakonec kořen ohodnocen hodnotou -1. Dokažte nebo vyvraťte každé z následujících tvrzení:

- Neexistuje partie (tj. cesta z kořene do listu), v níž by vyhrál bílý.
- Existuje vítězná strategie černého (tj. pro každý možný tah bílého má černý k dispozici protitah, který mu zajistí, že nakonec vždy vyhraje).