Binární vyhledávání

1. **Násobnost** V poli máme n vzestupně seřazených čísel $x_1, \ldots, x_n$. Jak upravit binární vyhledávání tak, aby vrátilo indexy prvního a posledního výskytu pro dané číslo k?

2. **Mezery** Máme posloupnost čísel $1, \ldots, n$ po sobě jdoucích čísel s k chybějícími čísly. Jak najdeme první chybějící číslo? A jak poslední?

3. **Nekonečné vyhledávání** Mějme neklesající funkci f přiřazující přirozeným číslům hodnoty 0 a 1. Jak efektivně najdeme nejmenší přirozené číslo m t.ž. $f(m) = 1$? Nemáme žádný hodní odhad na m. Chceme časovou složitost omezenou pomocí m.

4. **Rychlý odhad** Najděte horní odhad na m velikosti nejvýše $2m$ v čase $\mathcal{O}(loglog(m))$.

Práce s grafy

- Jaké jsou algoritmické reprezentace grafů?
- Jaké mají výhody a nevýhody?
- Jaká je 'ideální' reprezentace grafů?

5. **Jak poznat** - navrhněte řešení a odhadněte složitost
   - graf je bipartitní
   - graf je tripartitní
   - graf je souvislý
   - graf je strom
   - graf má trojúhelník

6. **BFS a cesty** - Jak upravíme BFS tak, aby pro vrchol v na vstupu a každý vrchol u dosažitelný z v spočítalo počet nejkratších cest z v do u?

7. **BFS a složitost** - Jaká je časová složitost BFS pokud máme k dispozici pouze matici sousednosti? Vyplatí se vytvořit z matice seznamy sousedů? Na čem závisí prostorová složitost DFS?

Domácí úkol

Kopec je posloupnost Y prvků $y_1, \ldots, y_k$ pro kterou existuje index $1 \leq i \leq k$ t.ž. $y_1, \ldots, y_i$ je neklesající a $y_i, \ldots, y_k$ je nerostoucí.

Mějme na vstupu posloupnost X nezáporných čísel $x_1, \ldots, x_n$. V posloupnosti X najděte největší kopec (podposloupnost největší délky, která je kopec).

Využijte datovou strukturu $\mathcal{S}$ s následujícími operacemi:

- $INIT()$ - inicializuje nebo vyprázdní strukturu, v čase $\mathcal{O}(1)$
- $INSERT(k, v)$ - vloží dvojici klíče k a hodnoty v, v čase $\mathcal{O}(log|\mathcal{S}|)$
- $QUERY(k)$ - $max\{v' | (v', k') \in \mathcal{S} \ \& \ k' \leq k\}$, v čase $\mathcal{O}(log|\mathcal{S}|)$

Kde $|\mathcal{S}|$ je počet dvojic v danou chvíli uložených v $\mathcal{S}$.

Domácí úkol by měl obsahovat:

- pseudokód a stručný (bez detailů) slovní popis algoritmu
- zdůvodnění korektnosti
- časová a prostorová složitost (pouze $\mathcal{O}$)
  - předpokládejte, že $\mathcal{S}$ vždy zabírá $\mathcal{O}(|\mathcal{S}|)$ paměti.
- Bude algoritmus fungovat i pokud povolíme záporná čísla?
- Jak si poradíme, když místo $\mathcal{S}$ dostaneme $\bar{\mathcal{S}}$ jejíž $QUERY$ bude vracet minimum místo maxima?

Bonusová otázka: Mluvili jsme hodně o binárním vyhledávání a jaké asymptotické rychlosti lze dosáhnout pro různé adaptace. Ukažte, že pro vyhledávání v seřazeném poli délky n je časová složitost $\mathcal{O}(logn)$ (v nejhorším případě) optimální. Tzn. dokažte následující implikaci pomocí formální definice asymptotické notace:

algoritmus $\mathcal{A}$ skončí v čase $o(logn)$ pro každý vstup $\rightarrow$ $\mathcal{A}$ není korektní.