
PAL: 13. cvičení

6. 1. 2022

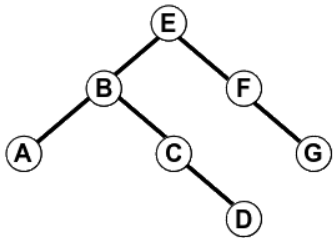
Př. 11/16: plněné B stromy

Je dán a) B-strom, b) B+ strom. Strom je řádu 10 a máme do něj umístit 100 000 klíčů. Jaký je maximální a minimální možný počet uzlů tohoto stromu? Jaká je maximální a minimální možná hloubka tohoto stromu?

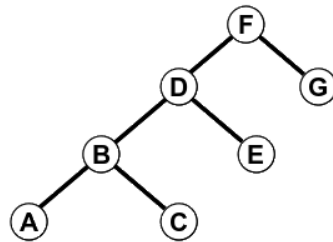
Př. 12/4: Červenočerné stromy

Navrhněte červenočerné obarvení daných stromů tak, aby vznikl korektní červenočerný strom. Prázdné (nil) listy nejsou zobrazeny.

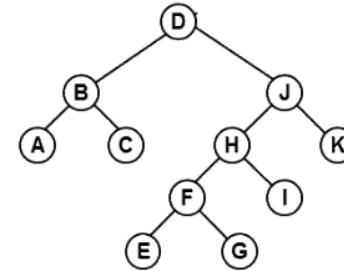
A.



B.



C.



Př. 13/10: k-d stromy

Sestavte k-d strom pro body v rovině $(50,50)$, $(20,35)$, $(60,15)$, $(30,40)$, $(50,40)$, $(40,50)$, $(15,60)$, $(15,35)$, $(35,40)$. Kolik možných stromů můžete dostat? Pokud byste strom nekonstruovali přímo, ale postupnými operace INSERT, jak bude vypadat výsledný strom? Bude se lišit od stromu zkonstruovaného přímo? Pro jaké setřídění bodů dostanete “nejlepší” strom? (A co je kritériem kvality stromu?) Pro jaké setřídění bodů dostanete naopak “nejhorší” strom?

Př. 13/13: k-d stromy

Ze stromu vzniklého v minulém příkladu postupnými operacemi INSERT odstraňte postupně body (50,50) a (20,35).

Př. 14/1: konstrukce trie

Vybudujte binární trie tak, že do něj postupně vložíte klíče 01010011, 00000111, 00100001, 01010001, 11101100, 00100001, 10010101, 01001010.

Př. 14/2: trie se 6 klíči

Nakreslete příklad binární trie, která obsahuje 6 klíčů, každý o 5 bitech. Trie bude obsahovat klíče 00000 a 11111, hloubka trie bude minimální možná.

Př. 14/3: paměťová stopa trie

Určete, kolik minimálně a kolik maximálně paměti je zapotřebí pro uložení binární trie obsahující 1.000.000 řetězců, z nichž každý má délku 50 ASCII znaků.

Př. 14/4: inverze klíčů

V daném binární trie chceme nahradit každý klíč jeho bitovým doplňkem. Navrhněte efektivní postup, který to provede bez budování nového trie a rušení starého. Jaká bude asymptotická složitost vaší metody?

Př. 14/5: slučování trie

Máme dva binární trie T1 a T2 a je zaručeno, že celkově obsahují unikátní klíče. Je možné sestavit metodu Merge(T1, T2), která spojí oba trie do jediného nového binárního trie a přitom nevytváří ani neruší žádné existující uzly v obou triích? Pokud ano, napište pseudokód.