

Rekurzivní definice délky lineárního seznamu (3 b)

odpoved:

Seznam má délku nula pokud je prázdný, jinak má délku jedna plus délka zbytku seznamu

Definice ekvivalence seznamu

odpoved:

Rekurzivní definice: Dva seznamy jsou ekvivalentní, když jsou oba prázdné a nebo když se rovnají jejich první prvky a současně jejich zbývajících částí seznamu.

Shell Sort , přepsat aby se nepřesouvaly hodnoty, ale indexy v poli

Por[] (8 b)

odpoved:

$$A[j] > A[j+step] \rightarrow A[Por[j]] > A[Por[j+step]]$$
$$A[j] := A[j+step] \rightarrow Por[j] := Por[j+step]$$

Popsat vkládání prvku před prvek, jednosměrný seznam (3 b)

odpoved:

Vložíme nový prvek metodou PostInsert za aktivní prvek. Vezmeme hodnotu aktivního prvku a zapíšeme ji do nového prvku. Hodnotu aktivního prvku přepíšeme novými daty a posuneme aktivitu (Succ)

Čím se liší BS se zpětnými ukazateli od normálního BS a implementace jakých operací se liší od implementace operací v normálním BS (4 b)

odpoved:

BS se zpětnými ukazateli používáme když se chceme při průchodu INORDER vyhnout rekurzi nebo použití zásobníku.

Zpětný uzel kořene ukazuje na NIL (všechny uzly vedlejší diagonály také)

Zpětný ukazatel levého syna ukazuje na svého otce.

Zpětný ukazatel pravého syna ukazuje tam kam otec.

Radix sort, čím se liší první průchod od ostatních (3 body)

odpoved:

V prvom cykle prechadza pole podľa indexu , v ostatných podľa ukazatelu

Infix to Postfix/Prefix nějakého výrazu. (6 bodů)

Infix -> PostFix:

odpoved:

$$(a+b)*(c-d)/(e+f)*(g-h) \Rightarrow ab+cd-*ef+/gh-*$$

Infix -> PreFix:

odpoved:

$$(a+b)*(c-d)/(e+f)*(g-h) \Rightarrow */*+ab-cd+ef-gh$$

TRP 2HASH + Brentova varianta (indexovane od 0) (6+6 b)

odpoved:

Bylo třeba doplnit číslo 59

i:	0	1	2	3	4	5	6	7	8	9	10
dat:	xx	xx	xx	xx	15			xx	19		
2HASH	xx	xx	xx	xx	15			xx	19		59
BRENT	xx	xx	xx	xx	59			xx	19		15

- Přečísľuju si (stačí si mi pamatovat dva vzorce) takže indexy jsou teď od 1..11
- Pro vložení X použiju vzorec: $(X \text{ Mod } \text{MAX}) + 1 = \text{indexPozice}$
- Pro posun X použiju vzorec: $(X \text{ Mod } (\text{MAX} - 1)) + 1 = \text{velikostKroku}$

Rekurzivní definice délky lineárního seznamu (4b ?)

odpoved:

Seznam má délku nula pokud je prázdný, jinak má délku jedna plus délka zbytku seznamu

Max. 30 slovy popsat rušení jednoho prvku v jednosměrném lineárním seznamu bez použití cyklů. (5b)

odpoved:

Je-li seznam jednoprvkový aktivita, první a poslední nastavím na NULL. Mažu-li první uložím si první, prvním se stane druhý a mažu uložený, Mažu-li poslední, ukazatel předposledního(aktualního) se ruší. Jinak ukládám ukazatel mazaného, mažu prvek a do ukazatele aktivního je načten uložený uk.

Shell Sort - přepsat tak, aby se nepřesouvaly hodnoty, ale indexy v

Max. 30 slovy popsat řešení vložení prvku v jednosměrném lineárním poli Por[] (6b ?)

odpoved:

$A[j] > A[j + \text{step}] \rightarrow A[\text{Por}[j]] > A[\text{Por}[j + \text{step}]]$
 $A[j] := A[j + \text{step}] \rightarrow \text{Por}[j] := \text{Por}[j + \text{step}]$

seznamu před aktivní (5b)

odpoved:

Vložím nový prvek metodou postInsert za aktivní prvek, vezmem hodnotu aktivního prvku a zapíšu ju do nového prvku, hodnotu aktivního prvku prepíšu novými datami a posuním aktivitu (succ)

Zápis s ukazatelem - DPP	Zápis s indexem - UDPP
Ptr je ukazatel	PtrI je index
nil	NilDMA (const NilDMA=0)
Ptr	PtrI
Ptr^	DMA.ArrItem[PtrI]
Ptr^.RPtr	DMA.ArrItem[PtrI].RPtr
Ptr^.RPtr^.LPtr	DMA.ArrItem[DMA.ArrItem[PtrI].RPtr].LPtr

5. dijskra na max 10 slov - funkce + použití (3b)

odpoved:

pouziva sa na binárne vyhľadavanie

-vychádza z predpokladu že moze byt viac poloziek s rovnakym klucom

-zabezpecuje stabilitu pri vkladani s bin. vyhľadáváním

3) Definice binárního stromu (2b)

odpoved:

BS je buď prázdný alebo pozostáva z 1 uzlu (koreňa) a 2 binarnych podstromov (Lavého a Pravého), oba tieto podstromy majú vlastnosti stromu.

Binární strom je buď prázdný, nebo sestává z jednoho uzlu zvaného kořen a dvou binárních podstromů - levého a pravého. (Oba podstromy mají vlastnosti stromu).

6) Popsat Sharovu metodu (3b)

odpoved:

Sharova metoda řeší případ, kdy skutečná velikost (počet prvků) tabulky je jiná, než je hodnota vhodná pro Uniformní binární nebo Fibonacciho vyhledávání. Metoda postupuje ve dvou krocích:

- V prvním kroku provede rozdělení na největším indexu, který vyhovuje metodě a který je menší než daná velikost.
- Ve druhém kroku zjišťuje, zde je hledaný klíč nalevo nebo napravo od dělicí hodnoty. Když je nalevo, postupuje jako by tabulka měla počet prvků daný rozdělovací (a pro metodu vhodnou) hodnotou. Když je napravo, provede transformaci tabulky posunem začátku pole doprava tak, aby prohledávaná část tabulky měla opět vyhovující počet prvků.

7) Rozdíl mezi 1. a dalšími průchody při radix sortu (3b)

odpoved:

V prvom cykle prechadza pole podľa indexu , v ostatných podľa ukazatel

11) Popsat, jak se z heapu dělá pole, podmínky, že je prvek terminální, jen s levým synem (3b)

odpoved:

nie som si istý:

1. najprv musí byť heap "preublany"
2. zoberieme najspodnejši-najpravejší a zameníme ho z korenom..
3. koren môžeme dať do pola
4. zasa "preublame" a pokračuje 2. krokom

preublani - znamená, že všetci otcovia majú pod sebou menších synov, robí sa to postupne od najspodnejšieho-najpravejšieho.

9.k čemu sa používa Dijkstra

odpoved:

používa sa na binárne vyhľadávanie

-vychádza z predpokladu, že v množine môže byť viac položiek s rovnakým kľúčom

10.Brentova varianta popisuje, kde sa používa

odpoved:

-je to varianta metódy TRP s dvoma rozptyľovacími funkciami.

-je vhodná za podmienky, keď počet prípadov úspešného vyhľadávania je častejší, než neúspešného vyhľadávania s následným vkladaním

- 3.ekvivalencia dvoch zoznamov (poučka)

odpoved:

Dva zoznamy sú ekvivalentné, ak sú oba prázdne, alebo sa rovnajú ich prvé prvky a súčasne ich zvyšky.

- 5.BVS so spätnymi pointermi (dany kód nejakého prechodu, bolo treba doplniť chýbajúce časti a vylúštiť, ako prechod sa tým algoritmom vykoná)

odpoved:

Niekde som čítal, že BVS so spätnymi pointermi má zmysel robiť len pri prechode INORDER, keď nechceme použiť rekurziu alebo frontu

- 7.ad. výškovo/váhuovo vyvážený strom definície.

odpoved:

*BS - je **váhuovo** vyvážený, keď pre všetky jeho uzly platí, že počet uzlov ľavého podstromu a praveho podstromu sa rovnajú alebo líšia o 1.*

*BS - je **výškovo** vyvážený, keď pre všetky jeho uzly platí, že výška ľavého podstromu a praveho podstromu sa rovná alebo sa líši o 1.*