

Vzorec vprašanj pri R1

Pri vsakem razdelku so še vprašanja:

- Razloži postopek s katerim sprogramiraš tole ... [namesto ... pridejo v poštev vse naloge z vaj]
- Vprašanja s <http://lokar.fmf.uni-lj.si/Vprasanja/PSiA>

SKLAD

1. S pomočjo primera razloži algoritem, s katerim preverjamo, ali so oklepaji v izrazu postavljeni pravilno. Oklepaji so seveda različnih vrst.
2. Podatke hranimo v skladu. Razloži, kako bi jih uredil, če je edina podatkovna struktura, ki je na voljo, sklad.
3. Opiši algoritem s pomočjo katerega bi določil globino gnezdenja oklepajev. Recimo v izrazu "[[][]] ()" je globina 2, v "[[][]()]" 1, v "[()]" spet 2, ...
4. Dana imamo dva razreda - Sklad1 in Sklad2. Prvi je implementacija sklada s tabelo, drugi pa s pomočjo enostavnega, enojno povezanega verižnega seznama. Opiši, kako bi bila videti funkcija, ki objekt mojSklad tipa Sklad1 pretvori v objekt tipa Sklad2. Torej če napišemo `s2 = pretvori(s1)` in je `s1` tipa Sklad1, je `s2` tipa Sklad2. Sveda oba vsebujeta enake podatke v enakem vrstnem redu!
5. Kako bi brez uporabe rekurzije sprogramiral hitro urejanje (Quick sort)
6. Razloži uporabo sklada pri izvajanju funkcij (v poljuben programskem jeziku).
7. Pri uporabi rekurzije ni tako redko, da dobimo sporočilo *"RecursionError: maximum recursion depth exceeded"* oziroma *"Process is terminated due to StackOverflowException."* Razloži, zakaj pride do tega!

VRSTA

1. Opiši, kako bi sestavil implementacijo vrste s pomočjo verižnega seznama.
2. Podatke hranimo v vrsti. Razloži, kako bi jih uredil, če za hranjenje podatkov lahko uporabljaš le vrste.
3. Kako bi sestavil predstavitev vrste s pomočjo enostavnega enojno povezanega verižnega seznama, če mora biti operacija vstavi $O(1)$, odstrani pa je lahko $O(n)$.
4. Opiši predstavitev APS vrsta
5. Opiši uporabo vrste pri pregledu grafa. Kaj se zgodi, če pri opisani uporabi vrsto zamenjamo s skladom?

O VS in verigah vozlov.

1. Kakšna je razlika med prazno verigo vozlov in praznim verižnim seznamom?
2. Dana imamo dva verižna seznama. V obeh so podatki urejeni nepadajoče. Sestavi algoritem, ki brez tvorbe novih vozlov sestavi nov verižni seznam, kjer so vsi podatki skupaj urejeni. SLIKA prej in SLIKA potem
3. Opiši, kako verižni seznam celih števil razdelimo na dva tri verižne seznama. V prvem so pozitivna, v drugem negativna števila. Elemente enake 0 spustimo. V VS pozitivnih naj bo relativni vrstni red enak kot prvoten, v VS negativnih pa naj bo ravno obraten. SLIKA prej

in SLIKA potem. Pri postopku smemo ustvariti le neko vnaprej določeno število vozlov. To število ne sme biti odvisno od števila vozlov prvotnega seznama.

4. Dan je verižni seznam, v katerem so podatku urejeni nenaraščajoče. Opiši algoritem s katerim odstraniš vse podvojene podatke.
5. Sestaviti moramo funkcijo, ki iz verižnega seznama odstrani sode elemente. Kakšna bo razlika med funkcijama, če je pri prvi parameter verižni seznam in pri drugi veriga vozlov.

Drevesa

1. Koliko je dvojiških dreves z n elementi, ki imajo enak premi pregled.
2. Opiši algoritem, s katerim vrneš vsa dvojiška drevesa, ki imajo določen vmesni in obratni pregled.
3. Sestavi primer para premi/vmesni pregleda dv. drevesa z vsaj petimi vozlišči, kjer je kljub temu, da se vsaj en podatek ponovi, le eno tako dv. drevo.
4. Kaj lahko poveš o dv. drevesih z različnimi podatki, ki imajo enak par pregledov (npr. enak vmesni in premi pregled ali pa enak obratni in premi pregled).

Vprašanja o časovni (in prostorski zahtevnosti).

1. Razloži, kako v $O(n)$ časa ugotovimo, katero strnjeno podzaporedje ima maksimalno vsoto.
2. Kako z merjenjem ugotovimo, ali je algoritem časovne zahtevnosti $O(nk)$ in kako določimo k .
3. Imamo algoritem časovne zahtevnosti $O(n)$. Ali je možno, da bi algoritem za izvajanje (na istem računalniku) za problem velikosti 100000 zahteval več časa kot algoritem $O(n^2)$? Utemelji!
4. Dana je množica celih števil. Analiziraj časovno zahtevnost vsaj dveh algoritmov za iskanje vseh parov celih števil, ki se seštejejo v 0.
5. Primerjaj časovno zahtevnost urejanja z mehurčki v najboljšem in najslabšem primeru.
6. Na različnih računalnikih izmerimo čase izvajanja algoritma za probleme velikosti 10, 100, 1000, 10000, 100000, 1000000, 10000000, 100000000, 200000000, 400000000, 800000000. Vsi nabori podatkov pri istih velikostih so enaki (za vse računalnike). Kako bi našli najhitrejši algoritem?
7. Kako bi med sabo primerjali različne algoritme, ki so napisani v različnih programskih jezikih? Algoritem lahko izvedemo na istem računalniku!
8. Trije algoritmi vsebujeta zanko v zanki. V notranji zanki opravimo konstantno mnogo operacij. Enako velja za dodatne operacije (torej tiste, ki jih opravimo poleg notranje zanke) v zunanji zanki. Izven zanke opravljeno delo je $O(1)$. Opiši, kako je možno, da je prvi algoritem $O(n^2)$, drugi $O(n)$ in tretji $O(n^3)$.
9. Analiziraj časovno in prostorsko zahtevnost računanja n -tega Fibonnacijevega števila, če uporabiš

- iterativen postopek
- rekurzivni postopek brez memoizacije
- rekurzivni postopek z memoizacijo

Pri ocenjevanju prostorske zahtevnosti uporabi dva različna modela:

- pri prvem štejejo tudi porabo prostora za klice funkcij
- pri drugi štejejo izključno prostor, potreben znotraj računanja posamezne funkcije

Dinamično programiranje

1. Opiši postopek reševanja 0/1 nahrbtnika, če nas zanima le optimalna vrednost.
2. Dane so vse potrebne množice S in Z . Določi, kako dosežemo optimalno polnitev za dani volumen.
3. Razloži, kaj vse lahko razberemo, če imamo dano množico S_{10} .
4. Razloži, na kakšen način iz S_i dobimo Z_{i+1} . Zakaj je, glede na definicijo Z_i in S_i to prav?
5. Denimo, da so vsi predmeti enake velikosti. Kako to vpliva na postopek reševanja 0/1 nahrbtnika?
6. Izpelji Bellmanovo enačbo za izračun optimalnega načina množenja zaporedja matrik
7. Denimo, da so vse matrike dimenzije $n \times n$. Se postopek določanja optimalnega načina množenja kaj poenostavi?
8. Denimo, da znamo matriki dimenzij $a \times b$ in $b \times c$ zmnožiti v času $a + 2b + c$. Kakšen bi bil postopek, ki določi optimalen način množenja
9. Denimo, da optimalni način določanja števila množenj izvedemo s postopkom memoizacije ustrezne Bellmanove enačbe. Skiciraj algoritem s pomočjo katerega bomo določili ustrezen način množenja.
10. Na konkretnem primeru 5 matrik izvedi postopek določanja optimalnega zaporedja množenj
11. Skiciraj algoritem, s katerim določiš katere med opravili, za katere poznaš plačilo, dan začetka in dan konca, boš opravil, da bo tvoj zaslužek čim večji. Vsako opravilo moraš opraviti v celoti, na posamezen dan smeš opravljati le enega.
12. Denimo, da imamo poljubno mnogo kovancev z vrednostjo d_1, d_2, \dots, d_n . Izpelji Bellmanovo enačbo, s katero določi število različnih načinov s katerim lahko točno plačaš znesek Z .