STU FIIT

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLÓGIÍ

Dátové štruktúry a algoritmy

Vyhodnotenie priebežného testu

15. 11. 2016

zimný semester 2016/2017

Úloha 1(A)

- (1 bod) Vysvetlite čo je to **asymptotický odhad zložitosti v priemernom prípade**, uveďte príklad nejakého štandardného algoritmu a zdôvodnite ho.

- Správne: taký odhad, ktorý uvažuje (rovnomerne) náhodný vstup veľkosti N, napr. Quicksort O(N log N) náhodný pivot celkom dobre rozdeľuje pole

- Nesprávne:
  - priemerne veľký vstup
  - Priemer medzi najlepším a najhorším prípadom
  - ...

Úloha 1(B)

- (1 bod) Uveďte **faktory ovplyvňujúce celkový čas behu programu** na konkrétnom počítači.

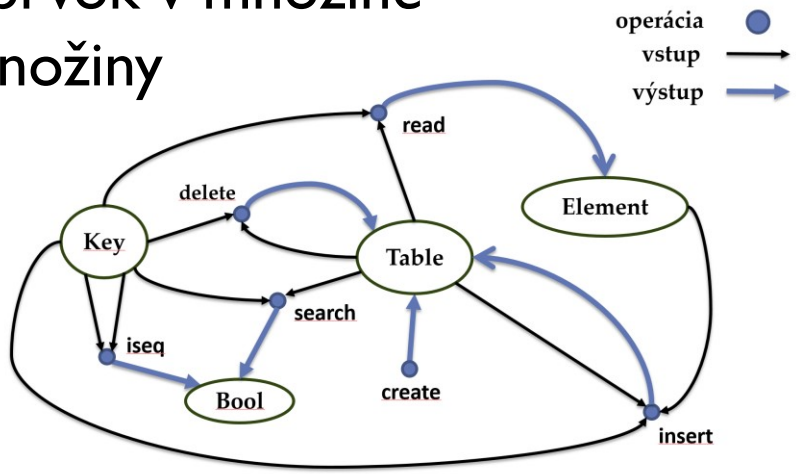
- Správne: také faktory, ktoré sú špecifické pre konkrétny počítač: procesor, pamäť (RAM, disky), stav OS

- Nesprávne: faktory, ktoré nezávisia na počítači:
  - Zložitosť je na všetkých počítačoch rovnaká...
  - Počet vykonaných inštrukcií detto..

Úloha 2(B)

- (2 body) Abstraktná dátová štruktúra (ADT): **dynamická množina**. Uveďte špecifikáciu operácií (nemusíte slovne opisovať):
- Správne:

  insert(x)– vložiť/pridať prvok do množiny
  element search(x) – vyhľadať prvok v množine
  delete(x) – odstrániť prvok z množiny
- Nesprávne:
  - Chýbajúce argumenty
  - Chýbajúce návratové hodnoty
  - Neúplné



Úloha 2(A)

- (2 bod) Abstraktná dátová štruktúra (ADT): **prioritný rad**. Uveďte špecifikáciu operácií (nemusíte slovne opisovať):
- Správne:

  insert(x)– vložiť prvok do množiny
  element deleteMax(x) – vrátiť najväčší / najmenší v množine a vymazať (prípadne ešte getMax)
- Nesprávne:
  - Chýbajúce argumenty
  - Chýbajúce návratové hodnoty
  - Neúplné

Úloha 3(A)

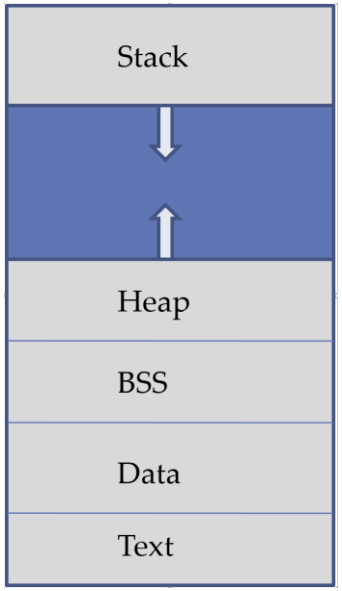
- (1 bod) Vysvetlite hlavné **rozdiely medzi procesom a programom**.

- Správne:
  - Program je statický kód a statické dáta
  - Proces je dynamická inštancia kódu, dát a ďalšieho
  - Bežiacemu procesu sa musí prideliť v počítači pamäť, aby mal kam zapisovať údaje (medzivýsledky atď.)
- Nesprávne:
  - „proces je vykonanie viac programov“,
  - „jeden program môže mať viacero procesov“

Úloha 3(B)

- (1 bod) Prideľovanie pamäti: vysvetlite **rozdiely medzi zásobníkom (stack) a haldou (heap)**.

- Správne:
  - **Stack (zásobník)**: lokálne premenné bežiaceho procesu
  - **Heap (halda voľnej pamäti)**:
    - dynamická pamäť procesu (môže sa zväčšovať aj zmenšovať)
    - pamäť, ktorú prideľuje malloc()



- Nesprávne:
  - Opisovať push/pop a haldu (insert/getMax) alebo rad (enqueue/dequeue)

Úloha 4(A)

- (3 body) Navrhnite algoritmus, ktorý **usporiada čísla N občianskych preukazov v čase O(N)**. Stručne opíšte hlavnú myšlienku a napíšte pseudokód.

- Správne: sú to reťazce konštantnej dĺžky, ideálny je radix sort – 8 prechodov.

- Nesprávne:
  - „použijem rozptýlenie hashovacou funkciou“
  - „použijem súčet ASCII znakov“
  - ...

Úloha 4(B)

- (3 body) Uvažujme množinu N reťazcov (rozličných dĺžok), napíšte pseudokód **usporadúvania týchto čísel využitím haldy** (operácie haldy nemusíte implementovať), zdôvodnite odhad výpočtovej zložitosti a vysvetlite či (áno/nie) je v tomto prípade toto usporadúvanie asymptoticky optimálny algoritmus usporadúvania porovnávaním.

- Správne:

```
for (i = 0; i < n; i++) insert(pq, str[i]);
for (i = 0; i < n; i++) result[i] = deleteMax(pq);
```

Zložitosť O(N log N * dĺžka najdlhšieho reťazca)

optimálne to asi nie je (radix je lepší)

Úloha 4(B)

- (3 body) Uvažujme množinu N reťazcov (rozličných dĺžok), napíšte pseudokód **usporadúvania týchto čísel využitím haldy** (operácie haldy nemusíte implementovať), zdôvodnite odhad výpočtovej zložitosti a vysvetlite či (áno/nie) je v tomto prípade toto usporadúvanie asymptoticky optimálny algoritmus usporadúvania porovnávaním.

- Nesprávne:
  - Perfektná implementácia heapify()
  - „Je to optimálne, lebo tie optimálne mám vždy problím napísať :)“
  - „Zložitosť haldy O(n)“
  - „Pred vložením do haldy reťazec zahashujem“

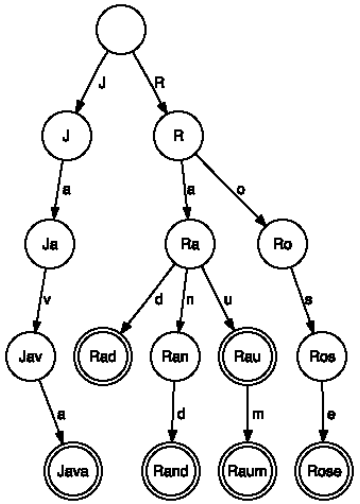
Úloha 5(A)

- (2 body) Vysvetlite **výhody B-stromu oproti červeno-čiernym stromom**, a uveďte situácie, v ktorých je vhodnejšie použitie B-stromov.

- Správne:
  - Má viac kľúčov vo vrchole a teda menšiu výšku
  - Výhodný pri veľkých množstvách údajov, pri prístupoch na disk (lebo ich bude málo), resp. efektívne načítanie do pamäte
- Nesprávne:
  - Hodí sa pre menšie počty
  - Vhodné ak nám záleží viac na poradí ako na čase spracovania
  - Že sa nevyvažuje, využíva mapovanie písmen
  - Že nie je potrebné prefarbovať (a vyvažovať)

Úloha 5(B)

- (2 body) **Do písmenkového stromu (trie) vkladám reťazec**. Vysvetlite ako bude prebiehať vkladanie a aká bude výpočtová zložitosť tejto operácie.



- Správne:
  - Vykonávanie operácií: začnem v koreni a postupujem po jednom písmenku a pridávam (ak ešte neexistujú) vrcholy do stromu.
  - Zložitosť O(n), kde n je dĺžka reťazca
- Nesprávne:
  - Abecedne porovnávam s písmenkom vo vrchole, ak je menšie idem doľava, inak doprava
  - Zložitosť: O(log n)
  - Detailny opis pointerov a polí...

Úloha 6(A)

- (3 body) Napíšte pseudokód operácie **in-order nasledovník prvku** (successor) v binárnom vyhľadávacom strome. Stručne opíšte hlavnú myšlienku.

- Správne: dva prípady (má pravé dieťa, nemá)

```
bintree TREE-SUCCESSOR(T):
    if RCHILD(T) <> nil
            then return TREE-MINIMUM(RCHILD(T))
    S ← PARENT(T)
    while S <> nil and T = RCHILD(S) do
            T←S
            S ← PARENT(T)
    return S
```

- Nesprávne: inorder prechod, len jeden z prípadov

Úloha 6(B)

- (3 body) Napíšte pseudokód operácie **in-order predchodca prvku** (predecessor) v binárnom vyhľadávacom strome. Stručne opíšte hlavnú myšlienku.

- Analogicky ako nasledovník.

Úloha 7 (A+B)

- (1 bod) Vysvetlite čo je to **faktor naplnenia α**, aké sú vhodné hodnoty **α** pre rôzne typy hashovania (lineárne skúšanie, dvojité rozptýlenie, reťazenie) a stručne opíšte prečo.

- Správne:
  - V tabuľke je N prvkov v M vedierkach **faktor naplnenia** $\alpha = N/M$ (priemerný počet prvkov vo vedierku)
  - Otvorená adresácia vhodné okolo 0.5
  - Reťazenie vhodné malá konštanta (napr. 1)
- Nesprávne:
  - „náhodné napĺňanie hodnotami“
  - Koľko je rôznych prvkov s rovnakými hodnotami
  - Koľko je počet kolízií na počet vstupov
  - Že to je distribúcia prvkov v poli v závislosti od hashovacej funkcie a veľkosti poľa ...
  - Na základe faktoru vieme pridať prvok
  - Rozdieľ poštu prvkov v ľavom podstrome a pravom podstrome

Úloha 8(A)

- (2 body) Uvažujme rodnú matriku – register obyvateľov SR podľa rodného čísla. Keď k referentovi príde nejaká požiadavka, základná operácia je vyhľadanie záznamu podľa rodného čísla. Navrhnite hashovanie pre túto situáciu: **navrhnite hashovaciu funkciu (uveďte pseudokód)**, **určite vhodnú veľkosť tabuľky** a **spôsob riešenia kolízií**. Stručne opíšte ako bude prebiehať vloženie prvku (narodenie), a zdôvodnite výpočtovú a pamäťovú zložitosť vášho riešenia.
- Správne: odhadnúť počet prvkov a správne aplikovať znalosti z predchádzajúcej úlohy
  - Počet obyvateľov cca 5 500 000, tak alebo otvorená adresácia a veľkosť tabuľky 10M, alebo reťazenie a veľkosť tabuľky 5M

Úloha 8(B)

- (2 body) Uvažujme našu fakultu, na študijné oddelenie prichádzajú študenti. Keď k referentovi príde nejaká požiadavka od študenta, základná operácia je vyhľadanie záznamu podľa mena a priezviska. Navrhnite hashovanie pre túto situáciu: **navrhnite hashovaciu funkciu (uveďte pseudokód)**, **určite vhodnú veľkosť tabuľky** a **spôsob riešenia kolízií**. Stručne opíšte ako bude prebiehať vloženie (zápis študenta), a zdôvodnite výpočtovú a pamäťovú zložitosť vášho riešenia.

- Správne: Analogicky ako v 8(A), tu je počet prvkov cca do 2000, čiže veľkosť tabuľky pri otvorenom adresovaní okolo 4000-5000, pri reťazení okolo 1000-2000.

Úloha 8(A+B)

- Nesprávne:
  - „obyvatelov je cca 5000000 tak navrhujem tabulku velkosti 5000“
  - „Hashodnota bude súčet cifier (rodné číslo cca max 30, potom reťazenie“
  - Študentov je 1200, tak hash tabuľku aspoň 5000, a hashovanie podľa prvého písmena (cca 30 hodnôt), navyše: vyhľadávam bubble sortom...
- Dobré riešenie: Ááá, dajme tam 8 000 000, aby bolo dosť miesta aj pre ďalších obyvateľov, lineárne skúšanie

Úloha 8(A+B)

- Od jedného extrému k druhému extrému:
- Minimalistický prístup:
  - Veľkosť tabuľky 10 pri **lineárnom skúšaní** (a bonus, hashfunkcia stále vracala jednu hodnotu – 0)
- Maximalistický prístup:
  - Veľkosť hash tabuľky 26^{26} a kolízie zreťazením!

Získané body