

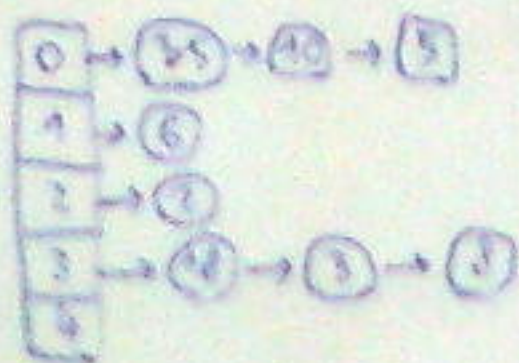
Podpíšte tento list aj dvojhárok – meno, priezvisko a osobné číslo.

Odpovede píšete priamo na tento list. Pracujete samostatne a odovzdáte len výsledky vlastnej práce, dosiahnute bez pomoci.

Meno a priezvisko:

osobné číslo:

1 – (3b) – Uvažujte prázdnu rozptylovú tabuľku, ktorej veľkosť je 5 (miest pamäti) a rozptylová funkcia je $h(x) = x \bmod 5$, pričom kolízie sa riešia zreťazením. Nakreslite náčrt stavu po vložení postupnosti prvkov (kľúčov) 35, 2, 18, 6, 3, 10, 8, 5. (Nekreslite priebežné stavy.)



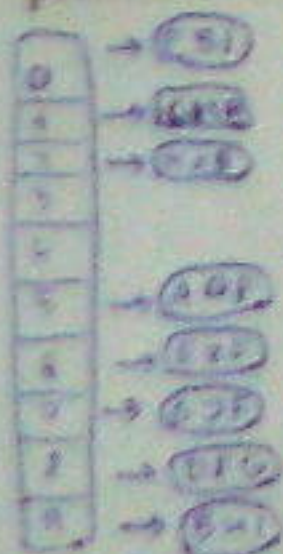
2 – (3b) – Uvažujte prázdnu rozptylovú tabuľku, ktorej veľkosť je 10 (miest pamäti) a rozptylová funkcia je $h(x) = x \bmod 10$, pričom kolízie sa riešia lineárnym skúšaním. Nakreslite náčrt stavu po vložení postupnosti prvkov (kľúčov) 35, 2, 18, 6, 3, 10, 8, 5. (Nekreslite priebežné stavy.)



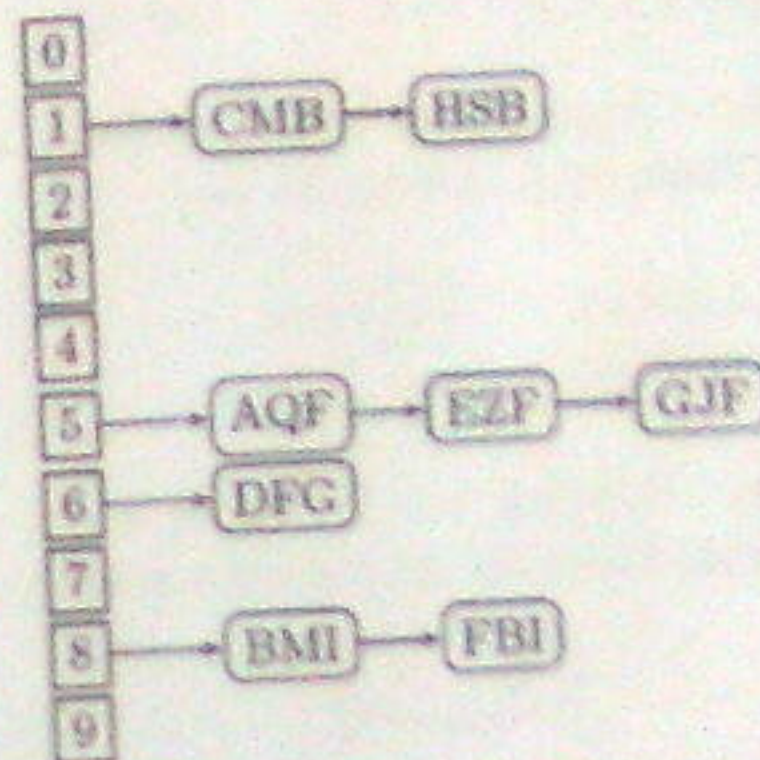
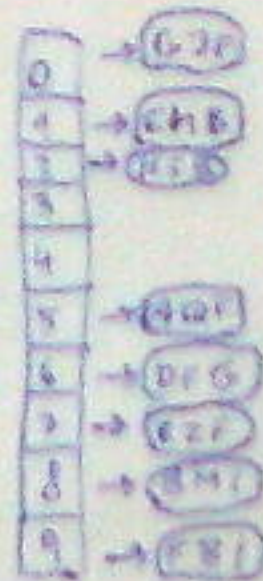
3 – (3b) – Uvažujte rozptylovú funkciu s riešením kolízií reťazením.

Ukážte, ako by vyzerala rozptylová tabuľka (na obrázku) s tými istými prvkami, ak by sa použilo na riešenie kolízií otvorené adresovanie s lineárnym skúšaním.

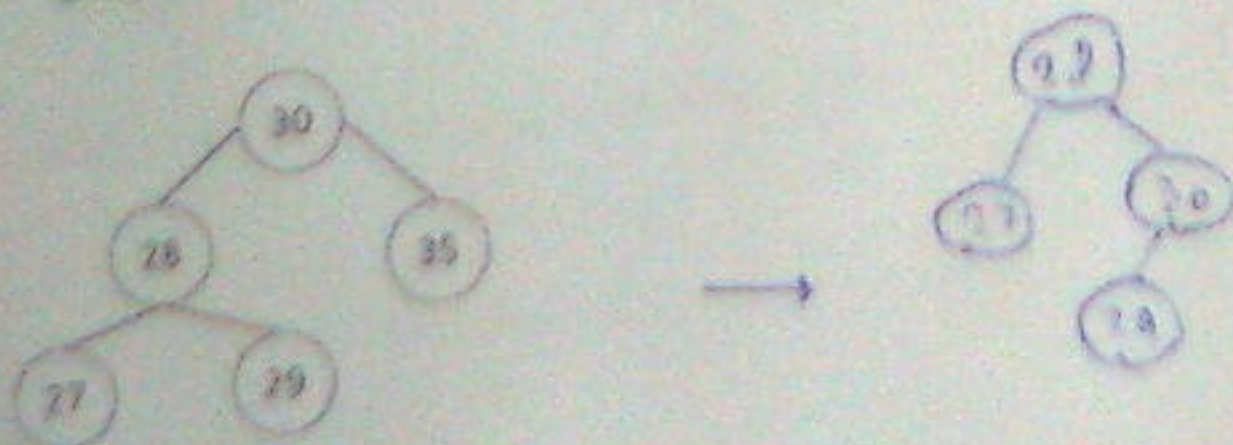
Je Vaše riešenie predchádzajúcej otázky jedinečné? Ak áno, napíšte prečo. Ak nie, napíšte prečo a nakreslite príklad iného riešenia.



Riešenie nie je dedukovateľné, pretože pri riešení kolízií otvoreným adresovaním s lin. skúšaním závisí kolokovanie elementov v tabuľke aj od poradia elementov pri vkladaní.



4 – (2b) – Uvažujte AVL strom na obrázku nižšie. Nakreslite strom po vymazaní prvku 35.



Podpíšte tento list aj dvojhárok – meno, priezvisko a osobné číslo.

Odpovede píšete priamo na tento list. Pracujete samostatne a odovzdáte len výsledky vlastnej práce, dosiahnuté bez pomoci.

Meno a priezvisko:

osobné číslo:

- ✓ 5 – (6b) – Napište algoritmus (môže byť v pseudokóde), ktorý vzostupne usporiada postupnosť celých čísel z intervalu (0, 1000) zapísaných v $A[1..n]$. Napište asymptotický odhad časovej zložitosti Vášho algoritmu vyjadrený pomocou počtu operácií sprístupnenia prvku poľa alebo pomocou nejakej inej realistickej metriky. Čím rýchlejší algoritmus napíšete, tým bude zaň viac bodov.

(odpoveď píšete na dvojhárok)

- ✓ 6 – (4b) – Napište algoritmus (môže byť v pseudokóde), ktorý z danej binárnej min-haldy nájde (vypíše) všetky hodnoty menšie než x .

(odpoveď píšete na dvojhárok)

- 7 – (2b) – Nech Σ^* označuje množinu všetkých možných reťazcov nad abecedou Σ . Čo znamená, že w je prípona reťazca x ? (zakrúžkujte vždy áno alebo nie)

Pre nejaké $y \in \Sigma^*$, $wy = x$	áno / <u>nie</u>
Pre nejaké $y \in \Sigma^*$, $yw = x$	<u>áno</u> / nie
Pre nejaké $y \in \Sigma$, $wy = x$	áno / <u>nie</u>
Pre nejaké $y \in \Sigma$, $yw = x$	<u>áno</u> / nie
Pre nejaké $y \in \Sigma^*$, wyx	áno / <u>nie</u>

- 8 – (2b) – Algoritmus Boyera a Moora na porovnávanie reťazcov porovnáva vzor s textom (zakrúžkujte vždy áno alebo nie)

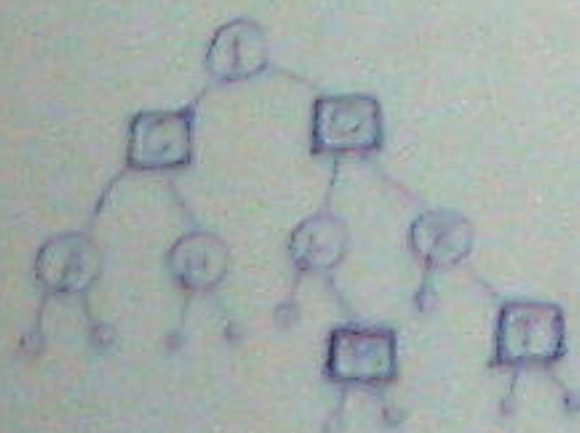
Zľava doprava	<u>áno</u> / nie
Sprava doľava	áno / <u>nie</u>

Zhora nadol	<u>áno</u> / nie
Zdola nahor	áno / <u>nie</u>

- 9 – (4b) – Nakreslite 2-3-4 strom, ktorý vznikne postupným vkladáním kľúčov A, L, G, O, R, I, T, H, M do stromu, ktorý bol na začiatku prázdny.



Nakreslite červeno-čierny strom, ktorý vznikne postupným vkladáním tých istých kľúčov do stromu, ktorý bol na začiatku prázdny.



LEGENDA:

- 1 ... červený uzel (internal)
- G ... červený uzel (leaf)
- O ... červený pomocný (null) uzel

Podpište tento list aj dvojhárok – meno, priezvisko a osobné číslo.

Odpovede píšete priamo na tento list. Pracujete samostatne a odovzdáte len výsledky vlastnej práce, dosiahnuté bez pomoci.

Meno a priezvisko:

osobné číslo:

10 – (6b) – 6 algoritmov usporadúvania. V najľavejšom stĺpci je vstupná postupnosť reťazcov, ktorá sa má usporiadať. V najpravejšom stĺpci je výsledná usporiadaná postupnosť. V ostatných stĺpcoch medzi nimi sú nejaké priebežné stavy usporadúvania jedným z 6 algoritmov uvedených nižšie. Pod každý z týchto stĺpcov napíšte písmeno príslušného algoritmu. Každé písmeno použite práve raz.

that	been	also	also	into	been	year	also
even	even	down	back	even	even	with	back
than	ever	come	been	than	from	will	been
been	fell	been	come	been	more	more	come
from	from	back	down	from	next	were	down
next	loss	even	even	next	over	plea	even
show	more	even	even	show	plea	well	ever
with	next	into	fell	jobs	show	lost	fell
more	over	fell	from	more	than	even	from
were	plea	from	have	much	that	some	have
over	show	jobs	into	over	were	very	into
plea	than	next	jobs	plea	with	next	jobs
fell	that	have	with	fell	fell	lead	lead
time	time	lead	time	back	time	time	loss
loss	were	loss	loss	loss	loss	that	lost
ever	with	over	show	ever	ever	jobs	more
lost	also	lost	lost	lost	lost	been	much
also	come	more	that	also	also	also	next
down	down	much	more	down	down	down	over
said	have	show	said	said	said	said	plea
some	lost	plea	some	some	some	from	said
have	said	that	were	have	have	have	show
very	some	said	very	lead	very	over	some
come	very	some	than	come	come	come	than
into	back	will	over	that	into	into	that
lead	into	very	lead	very	lead	fell	time
back	jobs	time	next	time	back	back	very
year	lead	than	year	year	year	than	well
will	much	with	will	will	will	show	were
well	well	well	well	well	well	loss	will
much	will	were	much	were	much	much	with
jobs	year	year	plea	with	jobs	ever	year

E F C A B D

- A. Shellovo usporadúvanie
 X Usporiadanie vkladáním (insert)
 X Rýchle usporadúvanie (bez znáhodnenia) (quick)
 D. Usporiadanie výberom (select)
 X Usporiadanie zlučováním (merge)
 X. Heapsort

11 – (4b) – Na obrázku (na nasledujúcej strane) je naznačená tabuľka všetkých možných tvarov 2-3-4 stromov, ktoré môžu vzniknúť vložením N rôznych kľúčov do stromu, ktorý bol na začiatku prázdny, pre N od 1 po 6. V ľavom stĺpci je počet kľúčov, v ďalšom stĺpci je počet možných rôznych tvarov stromov s toľkými kľúčmi a ďalej vpravo sú nakreslené tvary týchto stromov.

Dátové štruktúry a algoritmy

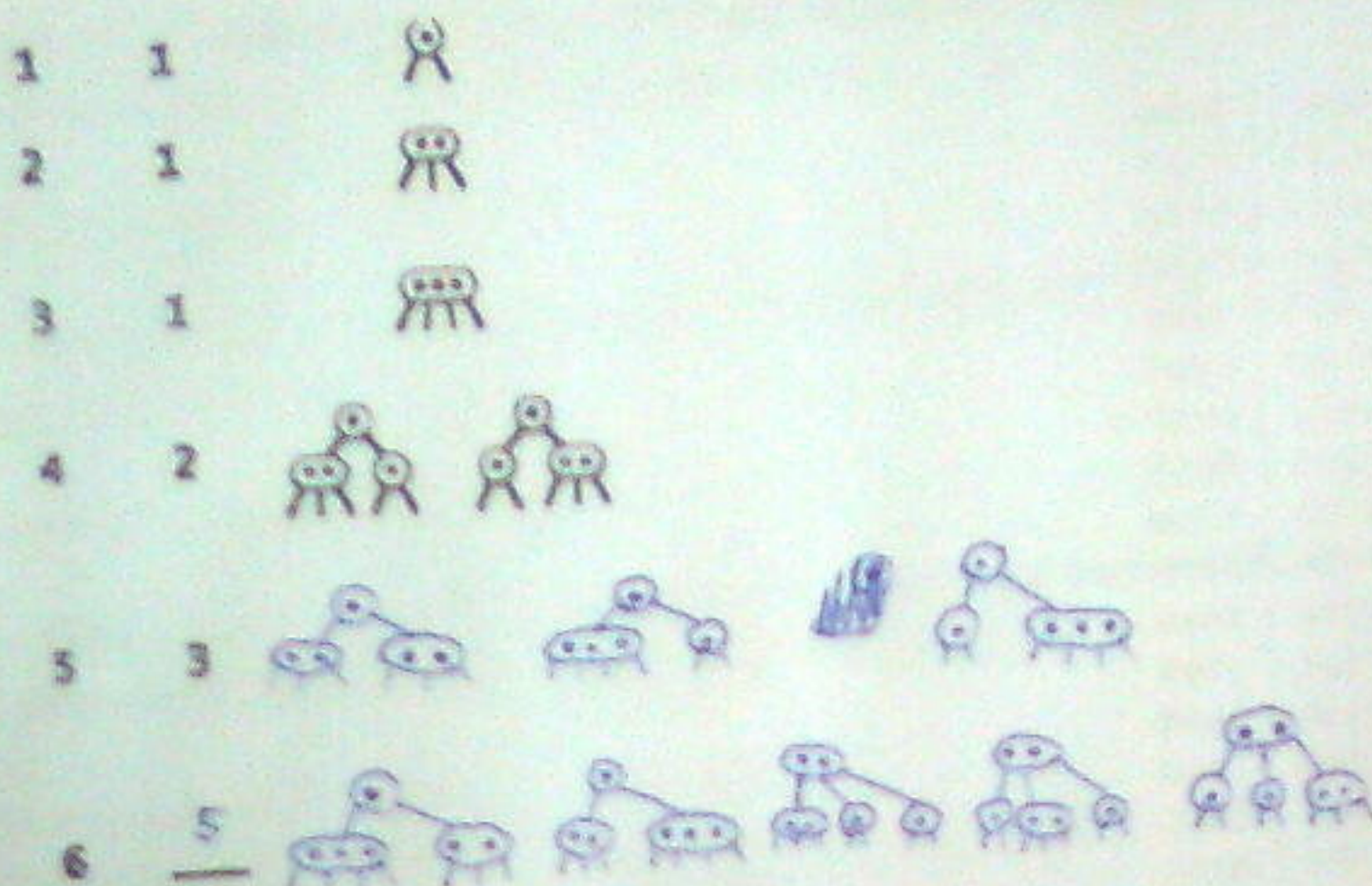
Podpište tento list aj dvojhárok – meno, priezvisko a osobné číslo.

Odpovede píšete priamo na tento list. Pracujete samostatne a odovzdáte len výsledky vlastnej práce, dosiahnuté bez pomoci.

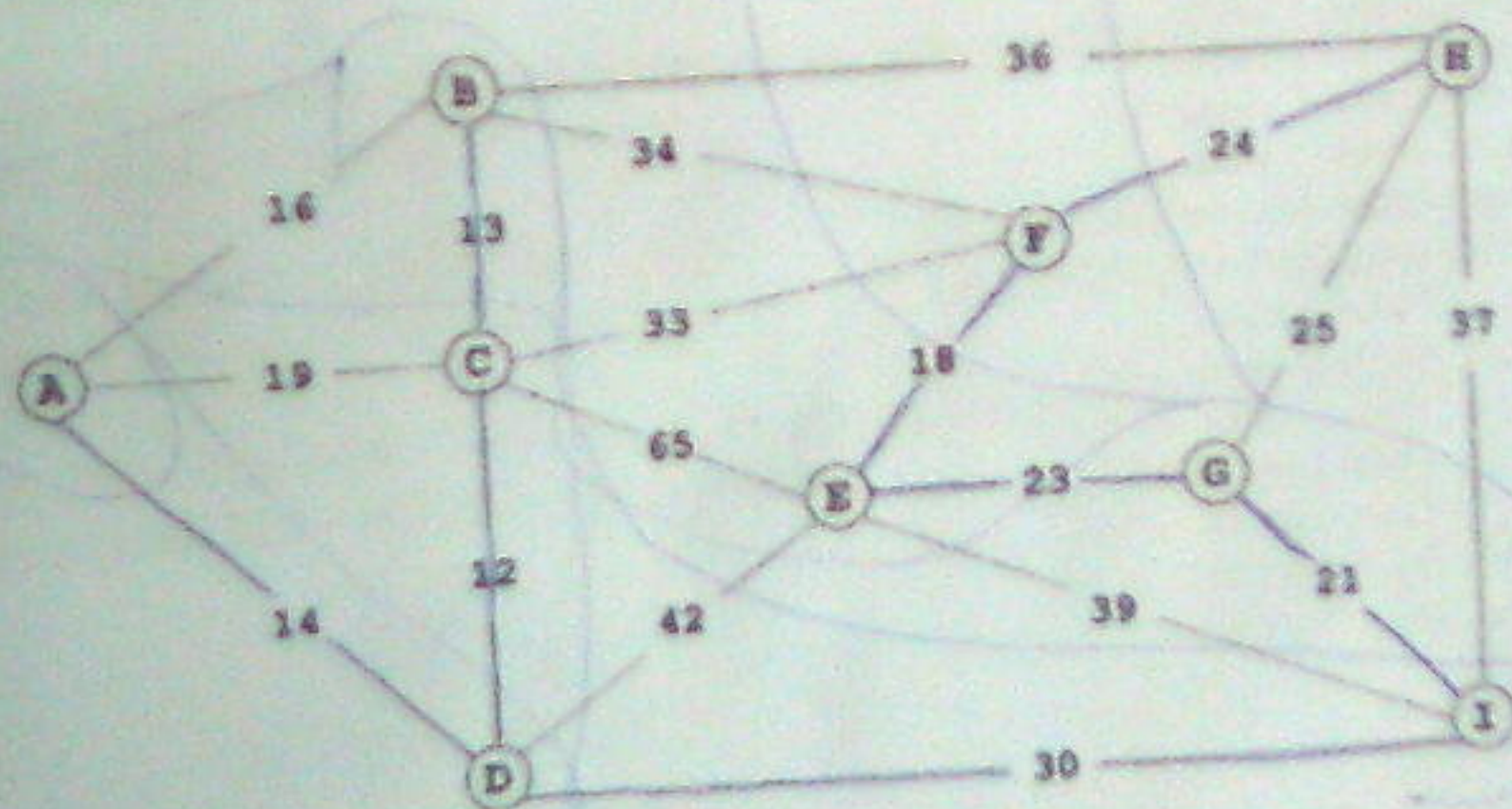
Meno a priezvisko:

osobné číslo:

Doplňte posledné dva riadky tabuľky (nakreslite všetky 3 možné tvary stromov s 5 kľúčmi a napíšte počet rôznych tvarov stromov s 6 kľúčmi a nakreslite ich).



12 – (6b) – Uvažujte hranovo ohodnotený neorientovaný graf:



Napište zoznam hrán v minimálnej kostre tohto grafu, ako ich bude do nej pridávať Kruskalov algoritmus. (Pomôcka: váhy hrán v stúpajúcom poradí sú: 12 13 14 16 18 19 21 23 24 25 30 33 34 36 37 39 42 65)

12, 13, 14, 16, 18, 19, 21, 23, 24, 25

30, 33, 34, 36, 37

Napište zoznam hrán v minimálnej kostre tohto grafu, ako ich bude do nej pridávať Primov algoritmus, ktorý začne od vrchola A.

14, 12, 13, 30, 21, 23, 24, 25

16, 18, 19, 39, 34, 36, 37

Podpište tento list aj dvojčiarok – meno, priezvisko a osobné číslo.

Odpovede píšete priamo na tento list. Pracujete samostatne a odovzdáte len výsledky vlastnej práce, dosiahnuté bez pomoci.

Meno a priezvisko:

osobné číslo:

Bonus – (10b)

Uvažujme usporiadanú množinu prvkov S . Binárna min-max halda T prvkov S je dátová štruktúra (binárny strom) s vlastnosťami ako bežná binárna halda avšak:

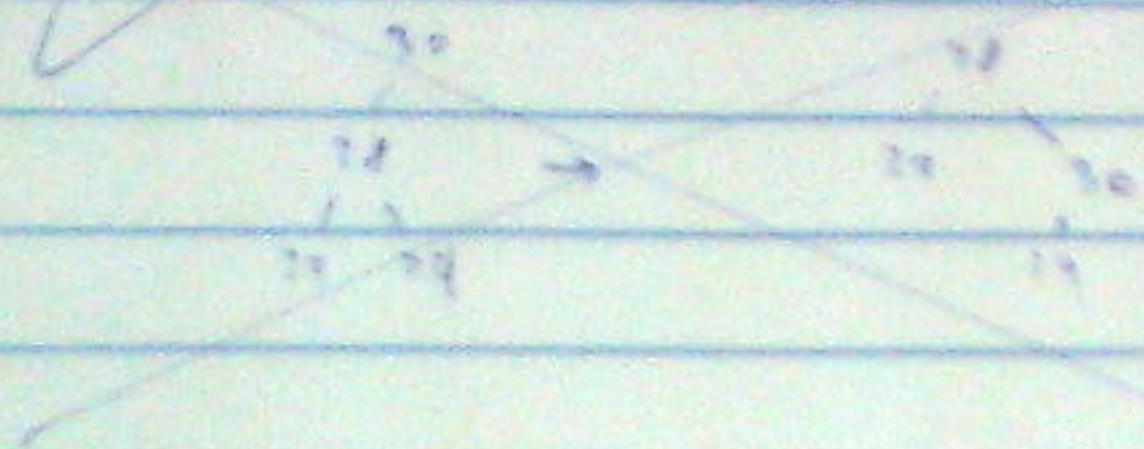
- T je min-max usporiadaná, t.j. hodnoty zapísané vo vrcholoch na párnej (nepárnej) úrovni sú menšie (väčšie) alebo rovné než hodnoty zapísané v ich nasledovníkoch (ak nejakých majú). Koreň je na úrovni 0. Takže najmenšia hodnota z množiny S je zapísaná v koreni T a najväčšia hodnota z množiny S je zapísaná v jednom z nasledovníkov koreňa.
- Min-max halda s n prvkami sa dá zapísať do vektora $A[1..n]$.

Min-max balda s n prvkami sa dá zapísať do vektora $A[1..n]$.

Implementujte operáciu vloženia prvku do min-max haldy.

$\text{insert}(\text{arr}, \text{len})$ {
 $\text{arr}[\text{len}]$
 $\text{size}(\text{arr}) = \text{size}(\text{arr}) + 1$;
 $\text{arr}[\text{size}(\text{arr})] = \text{elem}$;
}

④



⑤

BUCKET SORT (A, N) {

FOR $i = 1$ TO N DO {

 BUCKET $[F(A[i])]$ $\leftarrow A[i]$

}

FOR $i = 0$ TO N DO {

 INSERT SORT (BUCKET $[i]$);

}

RETURN;

$C \leftarrow$ BUCKET $[0] \dots$ BUCKET $[N]$ joined in order;

RETURN C;

}

$F(x)$ {

 RETURN $x \bmod 1000$;

}

~~ASYNCHRONOUS~~

ASYMPTOT. O(N) CAS. SLOTTOSTI:

$O(N)$ — ~~given~~

⑥

LOVEE_TRAY_X (HEAP, N) {

~~WHILE ((EXTRACT-MIN(HEAP)) \leq X) DO {~~

 PRINT (X);

}

}

EXTRACT-MIN (HEAP) {

~~LOCALLY~~ $y \leftarrow$ HEAP $[0]$;

 HEAP $[0] \leftarrow$ HEAP $[LAST(HEAP)]$;

 SIZE (HEAP) \leftarrow SIZE (HEAP) $- 1$;

 HEAPIFY-DOWN (HEAP, 0);

 RETURN y;

}

HEAPIFY-DOWN (HEAP, N) {

~~LOCALLY~~ $L \leftarrow$ LEFT (N);

~~LOCALLY~~ $R \leftarrow$ RIGHT (N);

~~WHILE~~

 IF $L \leq$ SIZE (HEAP) AND HEAP $[L] <$ HEAP $[N]$

 THEN LOWEST $\leftarrow L$;

 ELSE LOWEST $\leftarrow N$;

 IF $R \leq$ SIZE (HEAP) AND HEAP $[R] <$ HEAP $[LOWEST]$

 THEN LOWEST $\leftarrow R$;

~~RECURSIVE~~

 IF HEAP $[N] <$ HEAP $[LOWEST]$

 THEN EXCHANGE (HEAP $[N]$, HEAP $[LOWEST]$);

~~HEAPIFY-DOWN (HEAP, LOWEST)~~

 HEAPIFY-DOWN (HEAP, LOWEST);

}

LEFT (X) \leftarrow 2X;

RIGHT (X) \leftarrow (2X) $+ 1$;

←