

Универзитет у Новом Саду, Факултет техничких наука

ОАС Софтверско инжењерство и информационе технологије

Организација података

Статичка индекс-секвенцијална организација датотеке

Садржај

- **Увод**
- Структура
- Процедуре
- Одлике и примена
- Ресурси

- Статичка индекс-секвенцијална организација (Mogin, 2008)
 - статички карактер
 - непроменљиве димензије меморијског простора
 - присуство индекса
 - стабло намењено побољшању перформанси тражења случајно одабраног слога
 - примена секвенцијалне организације
 - основни слогови секвенцијално организовани
 - неки слогови могу бити и другачије организовани

Садржај

- Увод
- **Структура**
- Процедуре
- Одлике и примена
- Ресурси

Структура

- Сложена структура ^(Mogin, 2008)
 - три зоне
 - примарна зона (зона података)
 - основни слогови организовани по блоковима
 - зона индекса
 - стабло тражења
 - зона прекорачења
 - слогови прекорачиоци
 - три зоне могу бити посматране као три датотеке чији је садржај међусобно повезан показивачима

Структура

- Примарна зона (Mogin, 2008)
 - структура по узору на секвенцијалну датотеку
 - садржани основни слогови
 - слогови уређени према вредности кључа у растућем поретку
 - слогови груписани у блокове
 - B је број блокова у примарној зони ($B \geq 1$)
 - f^p је фактор блокирања у примарној зони
 - пожељно да f^p буде што веће
 - техничке карактеристике екстерног меморијског уређаја очекивано представљају ограничавајући чинилац у одређивању фактора блокирања
 - очекивано је коришћена ознака краја датотеке *
 - током рада над датотеком треба посебно водити рачуна о ознаци краја

Структура

- Примарна зона
 - пример

ПРИМАРНА ЗОНА

A_1^p	$\frac{5}{n(S_1)}$	$\frac{7}{n(S_2)}$	$\frac{10}{n(S_3)}$	A_2^p	$\frac{12}{n(S_4)}$	$\frac{14}{n(S_5)}$	$\frac{15}{n(S_6)}$
---------	--------------------	--------------------	---------------------	---------	---------------------	---------------------	---------------------

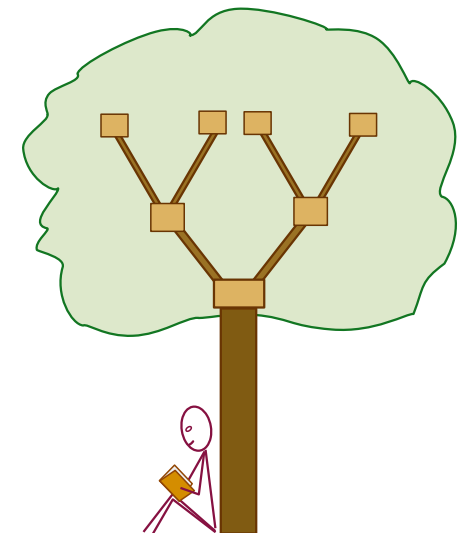
A_3^p	$\frac{19}{n(S_7)}$	$\frac{22}{n(S_8)}$	$\frac{26}{n(S_9)}$	A_4^p	$\frac{27}{n(S_{10})}$	$\frac{30}{n(S_{11})}$	$\frac{34}{n(S_{12})}$
---------	---------------------	---------------------	---------------------	---------	------------------------	------------------------	------------------------

A_5^p	$\frac{38}{n(S_{13})}$	$\frac{43}{n(S_{14})}$	$\frac{48}{n(S_{15})}$	A_6^p	$\frac{57}{n(S_{16})}$	$\frac{60}{n(S_{17})}$	$\frac{64}{n(S_{18})}$
---------	------------------------	------------------------	------------------------	---------	------------------------	------------------------	------------------------

A_7^p	$\frac{75}{n(S_{19})}$	*	
---------	------------------------	---	--

Структура

- Зона индекса (Mogin, 2008)
 - стабло тражења (стабло приступа)
 - пуно стабло реда n ($n \geq 2$) и висине h
 - ред стабла тражења n у пракси је број реда 50 до 100



Структура

- Зона индекса ^(Mogin, 2008)
 - стабло тражења
 - основна структура чвора стабла
 - садржано од 1 до n елемената
 - елемент је пар (k_e, A_e) , при чему $e \in \{1, \dots, n\}$
 - k_e је вредност кључа која се тиче слогова
 - A_e је адреса, при чему постоје две могућности
 - A_e је адреса блока примарне зоне који одговара k_e
 - A_e је адреса чвора у стаблу који садржи k_e
 - елементи су уређени према k_e у растућем поретку
 - две опште могућности за формирање елемента (k_e, A_e)
 - **пренос највећих вредности**
 - **пренос најмањих вредности**

пример структуре
чвора

$$A_c^i \begin{array}{|c|c|c|} \hline \frac{k_1}{A_1} & \dots & \frac{k_m}{A_m} \\ \hline \end{array}$$

Структура

- Зона индекса ^(Mogin, 2008)
 - стабло тражења – чворови
 - лисни чворови (листови)
 - лист који садржи m елемената ($1 \leq m \leq n$) покрива m блокова примарне зоне
 - сваком од покривених блокова примарне зоне одговара по један елемент
 - за сваки блок примарне зоне очекивано је да постоји одговарајући елемент у неком од листова стабла тражења
 - адреса A_e из елемента (k_e, A_e) јесте адреса блока у примарној зони

Структура

- Зона индекса ^(Mogin, 2008)
 - стабло тражења – чворови
 - лисни чворови (листови)
 - елементи листа садрже екстремне вредности кључа које се тичу блокова примарне зоне
 - две опште могућности за формирање елемента (k_e , A_e)
 - **пренос највећих вредности** – k_e је највећа вредност кључа заступљена у блоку примарне зоне A_e , осим ако је A_e адреса последњег блока примарне зоне, што подразумева коришћење највеће дозвољене вредности кључа
 - **пренос најмањих вредности** – k_e је најмања вредност кључа заступљена у блоку примарне зоне A_e , осим ако је A_e адреса првог блока примарне зоне, што подразумева коришћење најмање дозвољене вредности кључа

Структура

- Зона индекса ^(Mogin, 2008)
 - стабло тражења – чворови
 - унутрашњи чворови
 - унутрашњи чвор који садржи m елемената ($1 \leq m \leq n$) има m директно подређених чворова
 - сваком директно подређеном чвору одговара по један елемент
 - адреса A_e из елемента (k_e, A_e) представља адресу директно подређеног чвора у којем се налази елемент с вредношћу кључа k_e

Структура

- Зона индекса ^(Mogin, 2008)
 - стабло тражења – чворови
 - унутрашњи чворови
 - елементи унутрашњег чвора садрже по једну вредност кључа из сваког директно подређеног чвора
 - две опште могућности за формирање елемента (k_e, A_e)
 - **пренос највећих вредности** – k_e је највећа вредност кључа у чвору на адреси A_e
 - **пренос најмањих вредности** – k_e је најмања вредност кључа у чвору на адреси A_e

Структура

- Зона индекса

- пример

- случај преноса највећих вредности

ПРИМАРНА ЗОНА

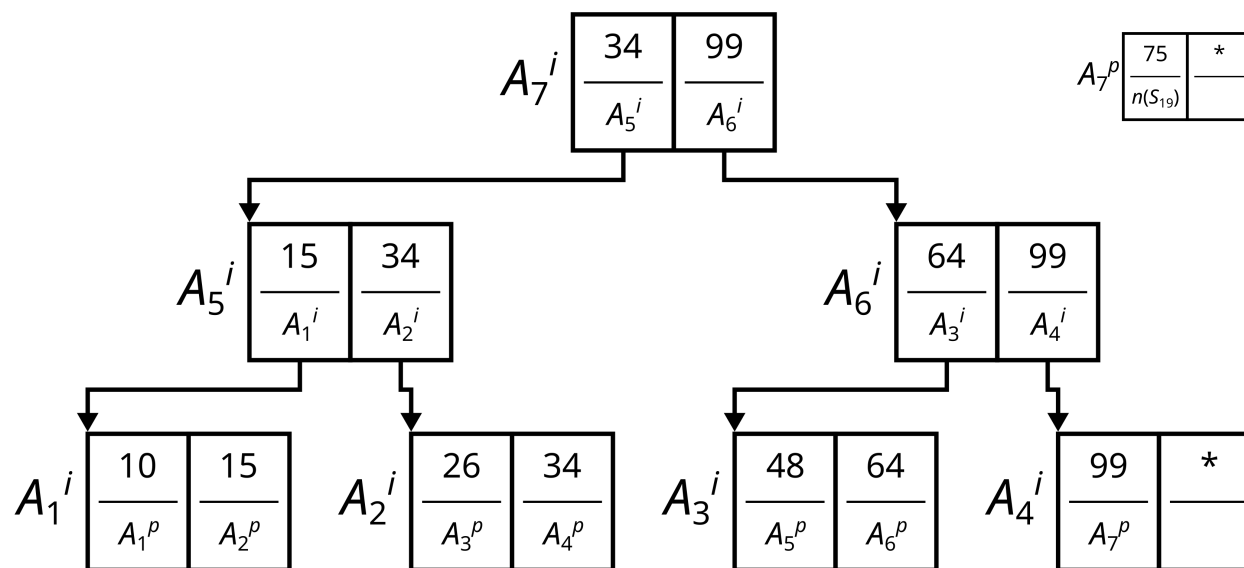
A_1^p	$\frac{5}{n(S_1)}$	$\frac{7}{n(S_2)}$	$\frac{10}{n(S_3)}$	A_2^p	$\frac{12}{n(S_4)}$	$\frac{14}{n(S_5)}$	$\frac{15}{n(S_6)}$
---------	--------------------	--------------------	---------------------	---------	---------------------	---------------------	---------------------

A_3^p	$\frac{19}{n(S_7)}$	$\frac{22}{n(S_8)}$	$\frac{26}{n(S_9)}$	A_4^p	$\frac{27}{n(S_{10})}$	$\frac{30}{n(S_{11})}$	$\frac{34}{n(S_{12})}$
---------	---------------------	---------------------	---------------------	---------	------------------------	------------------------	------------------------

A_5^p	$\frac{38}{n(S_{13})}$	$\frac{43}{n(S_{14})}$	$\frac{48}{n(S_{15})}$	A_6^p	$\frac{57}{n(S_{16})}$	$\frac{60}{n(S_{17})}$	$\frac{64}{n(S_{18})}$
---------	------------------------	------------------------	------------------------	---------	------------------------	------------------------	------------------------

A_7^p	$\frac{75}{n(S_{19})}$	*	
---------	------------------------	---	--

ЗОНА ИНДЕКСА



Структура

- Зона индекса

- пример

- случај преноса највећих вредности

ПРИМАРНА ЗОНА

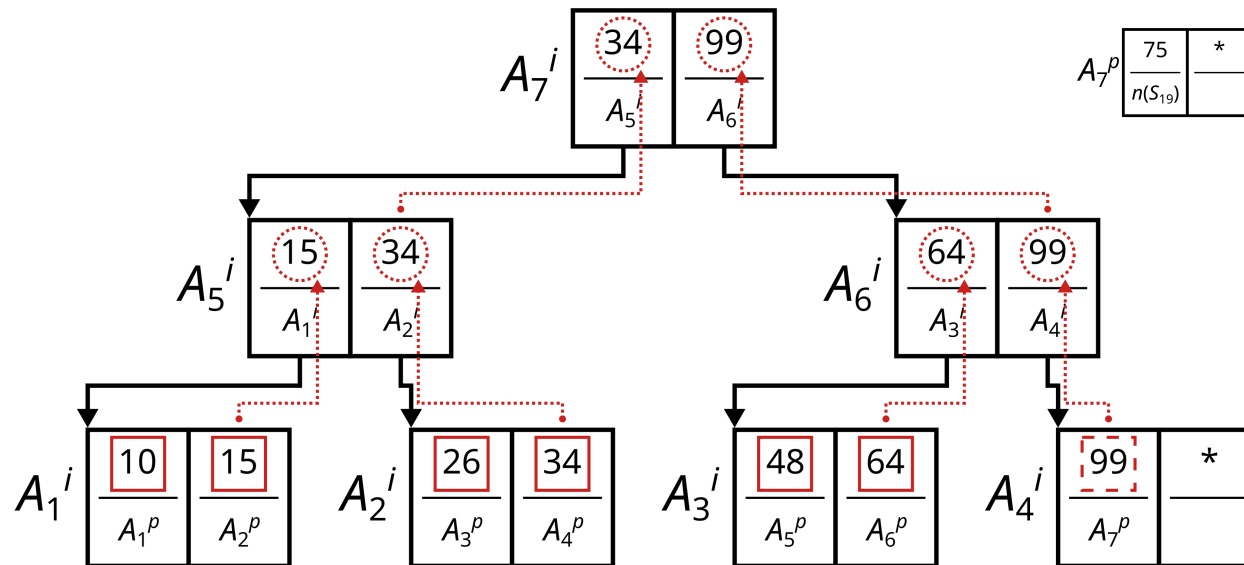
A_1^p	$\frac{5}{n(S_1)}$	$\frac{7}{n(S_2)}$	$\frac{10}{n(S_3)}$	A_2^p	$\frac{12}{n(S_4)}$	$\frac{14}{n(S_5)}$	$\frac{15}{n(S_6)}$
---------	--------------------	--------------------	---------------------	---------	---------------------	---------------------	---------------------

A_3^p	$\frac{19}{n(S_7)}$	$\frac{22}{n(S_8)}$	$\frac{26}{n(S_9)}$	A_4^p	$\frac{27}{n(S_{10})}$	$\frac{30}{n(S_{11})}$	$\frac{34}{n(S_{12})}$
---------	---------------------	---------------------	---------------------	---------	------------------------	------------------------	------------------------

A_5^p	$\frac{38}{n(S_{13})}$	$\frac{43}{n(S_{14})}$	$\frac{48}{n(S_{15})}$	A_6^p	$\frac{57}{n(S_{16})}$	$\frac{60}{n(S_{17})}$	$\frac{64}{n(S_{18})}$
---------	------------------------	------------------------	------------------------	---------	------------------------	------------------------	------------------------

A_7^p	$\frac{75}{n(S_{19})}$	*	
---------	------------------------	---	--

ЗОНА ИНДЕКСА



Структура

- Зона индекса

- пример

- случај преноса најмањих вредности

ПРИМАРНА ЗОНА

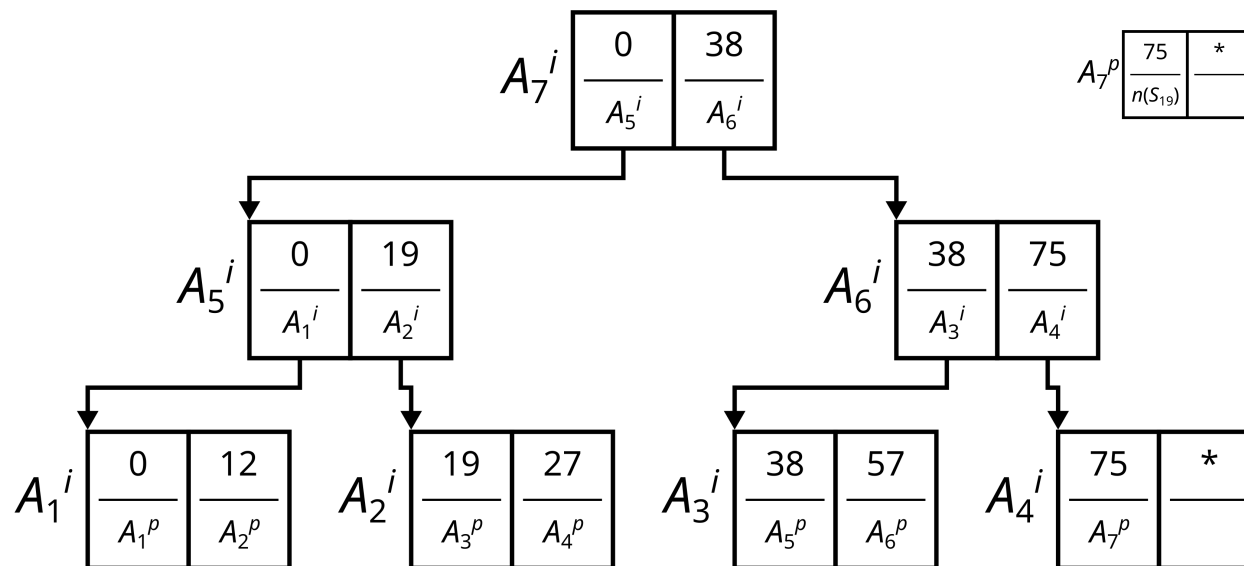
A_1^p	$\frac{5}{n(S_1)}$	$\frac{7}{n(S_2)}$	$\frac{10}{n(S_3)}$	A_2^p	$\frac{12}{n(S_4)}$	$\frac{14}{n(S_5)}$	$\frac{15}{n(S_6)}$
---------	--------------------	--------------------	---------------------	---------	---------------------	---------------------	---------------------

A_3^p	$\frac{19}{n(S_7)}$	$\frac{22}{n(S_8)}$	$\frac{26}{n(S_9)}$	A_4^p	$\frac{27}{n(S_{10})}$	$\frac{30}{n(S_{11})}$	$\frac{34}{n(S_{12})}$
---------	---------------------	---------------------	---------------------	---------	------------------------	------------------------	------------------------

A_5^p	$\frac{38}{n(S_{13})}$	$\frac{43}{n(S_{14})}$	$\frac{48}{n(S_{15})}$	A_6^p	$\frac{57}{n(S_{16})}$	$\frac{60}{n(S_{17})}$	$\frac{64}{n(S_{18})}$
---------	------------------------	------------------------	------------------------	---------	------------------------	------------------------	------------------------

A_7^p	$\frac{75}{n(S_{19})}$	*	
---------	------------------------	---	--

ЗОНА ИНДЕКСА



Структура

- Зона индекса

- пример

- случај преноса најмањих вредности

ПРИМАРНА ЗОНА

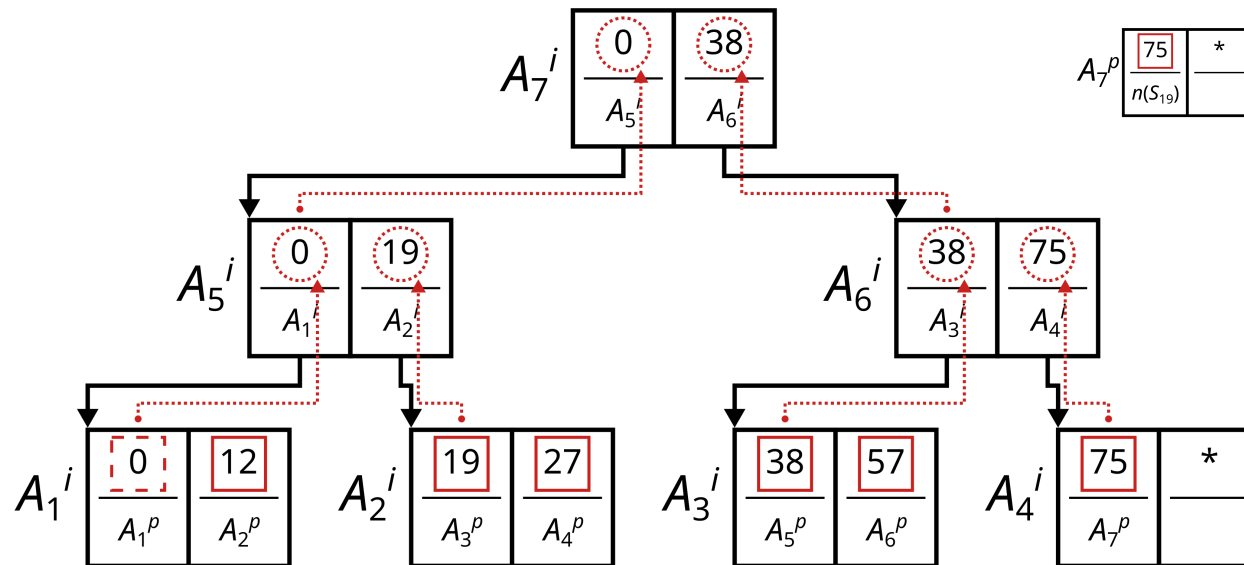
A_1^p	$\frac{5}{n(S_1)}$	$\frac{7}{n(S_2)}$	$\frac{10}{n(S_3)}$	A_2^p	$\frac{12}{n(S_4)}$	$\frac{14}{n(S_5)}$	$\frac{15}{n(S_6)}$
---------	--------------------	--------------------	---------------------	---------	---------------------	---------------------	---------------------

A_3^p	$\frac{19}{n(S_7)}$	$\frac{22}{n(S_8)}$	$\frac{26}{n(S_9)}$	A_4^p	$\frac{27}{n(S_{10})}$	$\frac{30}{n(S_{11})}$	$\frac{34}{n(S_{12})}$
---------	---------------------	---------------------	---------------------	---------	------------------------	------------------------	------------------------

A_5^p	$\frac{38}{n(S_{13})}$	$\frac{43}{n(S_{14})}$	$\frac{48}{n(S_{15})}$	A_6^p	$\frac{57}{n(S_{16})}$	$\frac{60}{n(S_{17})}$	$\frac{64}{n(S_{18})}$
---------	------------------------	------------------------	------------------------	---------	------------------------	------------------------	------------------------

A_7^p	$\frac{75}{n(S_{19})}$	*	
---------	------------------------	---	--

ЗОНА ИНДЕКСА

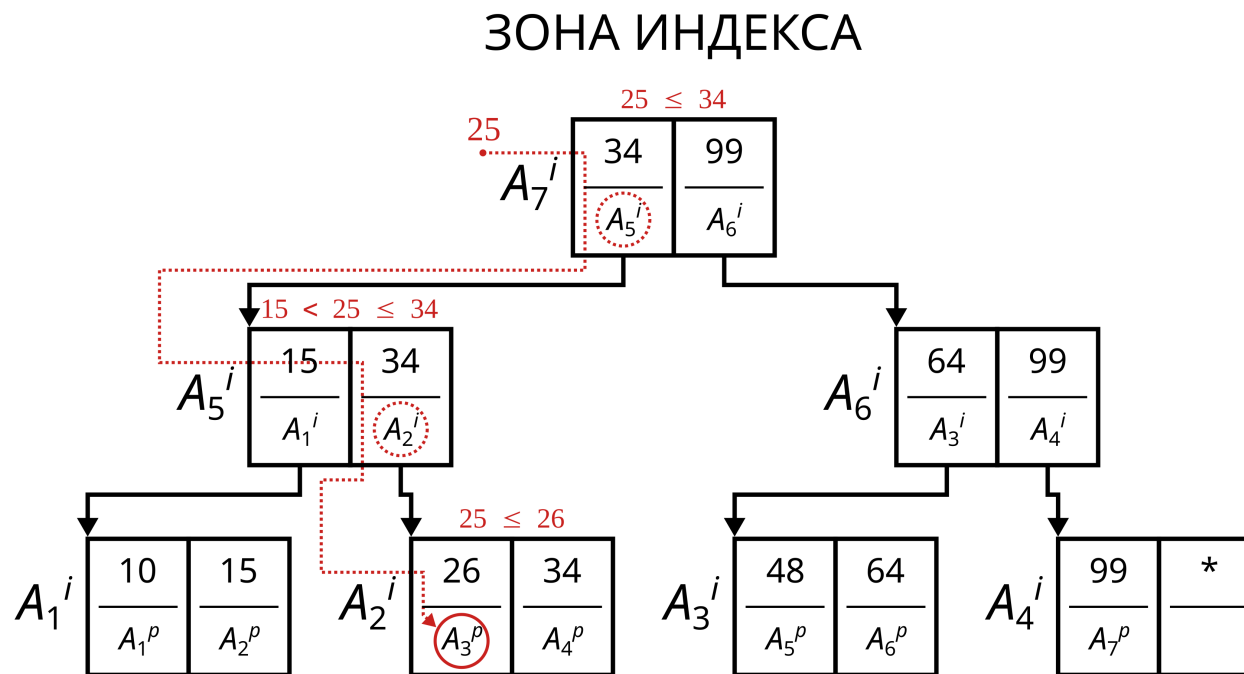


Структура

- Зона индекса ^(Mogin, 2008)
 - основни поступак тражења адресе у стаблу тражења
 - налажење адресе блока примарне зоне за вредност кључа k
 - налажење креће од корена као првог текућег чвора, а за сваки текући чвор издваја се адреса и по потреби задаје нови текући чвор
 - ако је текући чвор унутрашњи, адреса одговара новом текућем чвору
 - ако је текући чвор лисни, адреса одговара блоку примарне зоне
 - поступак издвајања адресе за текући чвор, чији су елементи $(k_1, A_1), \dots, (k_{e-1}, A_{e-1}), (k_e, A_e), (k_{e+1}, A_{e+1}), \dots, (k_m, A_m)$ за $1 \leq m \leq n$
 - случај преноса највећих вредности
 - ако $k \leq k_1$, адреса је A_1
 - ако $k_{e-1} < k \leq k_e$ где $e \in \{2, \dots, m\}$, адреса је A_e
 - случај преноса најмањих вредности
 - ако $k_e \leq k < k_{e+1}$ где $e \in \{1, \dots, m-1\}$, адреса је A_e
 - ако $k \geq k_m$, адреса је A_m

Структура

- Зона индекса
 - основни поступак тражења адресе у стаблу тражења
 - пример за вредност кључа $k = 25$ (случај преноса највећих вредности)



Структура

- Зона индекса ^(Mogin, 2008)

- стабло тражења – димензије

- висина стабла h

$$h = \lceil \log_n B \rceil$$

- број чворова на i -том нивоу хијерархије стабла C_i

$$C_i = \left\lceil \frac{B}{n^{h-i+1}} \right\rceil$$

- укупни број чворова стабла C

$$C = \sum_{i=1}^h \left\lceil \frac{B}{n^{h-i+1}} \right\rceil$$

- капацитет стабла $K = nC$

- укупни број елемената у стаблу E

$$E = \sum_{i=1}^h \left\lceil \frac{B}{n^{h-i}} \right\rceil$$

Структура

- Зона индекса ^(Mogin, 2008)
 - стабло тражења – намена
 - стабло тражења је ретко попуњени индекс
 - у стаблу је за сваки блок примарне зоне садржана само једна вредност кључа
 - стабло тражења представља табеларно задату бијективну функцију која пресликава скуп екстремних вредности кључа блокова примарне зоне на скуп адреса блокова
 - основна сврха је побољшање перформанси при приступању блоковима примарне зоне у случају тражења случајно одабраног слога

Структура

- Зона прекорачења (Mogin, 2008)
 - садржани слогови прекорачиоци
 - сваки блок примарне зоне може имати своје прекорачиоце, који су у оквиру зоне прекорачења спрегнути у листу
 - зона прекорачења садржи скуп једноструко спрегнутих листа прекорачилаца
 - димензије могу бити пројектоване у односу на примарну зону
 - нпр. капацитет зоне прекорачења одговара 15% капацитета примарне зоне

Структура

- Зона прекорачења (Mogin, 2008)
 - структура
 - индекс слободних локација L
 - евиденција о првој слободној локацији у зони прекорачења, али не за динамичко управљање слободним простором
 - слободне локације нису међусобно спрегнуте
 - блокови за смештање слогова прекорачилаца
 - фактор блокирања у зони прекорачења f^z често је 1
 - очекивано је мала вероватноћа да два логички суседна слога буду у суседним локацијама зоне прекорачења
 - употреба мањег блока подразумева бржу размену података између оперативне меморије и екстерног меморијског уређаја и мањи простор за смештај таквог блока у оперативној меморији
 - постоје поља показивача за потребе спрезања прекорачилаца

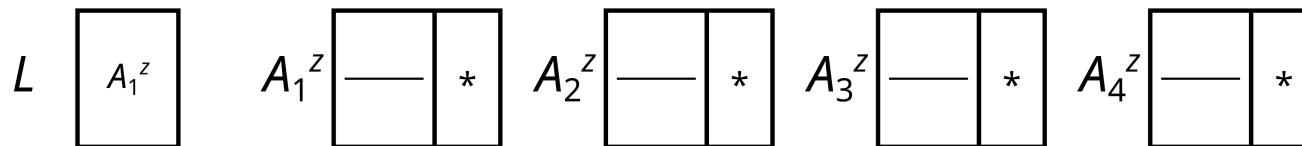
Структура

- Зона прекорачења

- пример

- случај када нема прекорачилаца (празна зона прекорачења)

ЗОНА ПРЕКОРАЧЕЊА



Структура

- Зона прекорачења ^(Mogin, 2008)
 - упис новог слога у датотеку може довести до уписа слога у зону прекорачења
 - за нови слог S с вредношћу кључа $k(S)$ употребом стабла тражења бива добијена адреса одговарајућег блока A_e
 - случај када блок јесте комплетан
 - $k_{e(max)}$ је тренутно највећа вредност кључа у том блоку
 - ако $k(S) < k_{e(max)}$
 - нови слог S треба да буде уписан у блок а тамошњи слогови с вредношћу кључа већом од $k(S)$ померени за једну локацију надесно
 - слог с вредношћу кључа $k_{e(max)}$ треба да буде уписан у зону прекорачења
 - ако $k(S) > k_{e(max)}$
 - нови слог S треба да буде уписан у зону прекорачења

Структура

- Зона прекорачења (Mogin, 2008)
 - упис новог слога у датотеку може довести до уписа слога у зону прекорачења
 - за нови слог S с вредношћу кључа $k(S)$ употребом стабла тражења бива добијена адреса одговарајућег блока A_e
 - случај када блок није комплетан
 - нови слог S треба да буде уписан у блок а тамошњи слогови с вредношћу кључа већом од $k(S)$ померени за једну локацију надесно

Структура

- Зона прекорачења (Mogin, 2008)
 - повезивање прекорачилаца
 - логички непосредно суседни прекорачиоци који се односе на исти блок примарне зоне спрегнути су у заједничку листу помоћу показивача
 - показивачи носе информацију о везама између слогова у логичкој структури података датотеке
 - основни поступци повезивања прекорачилаца са стаблом тражења
 - **директно повезивање**
 - **индиректно повезивање**

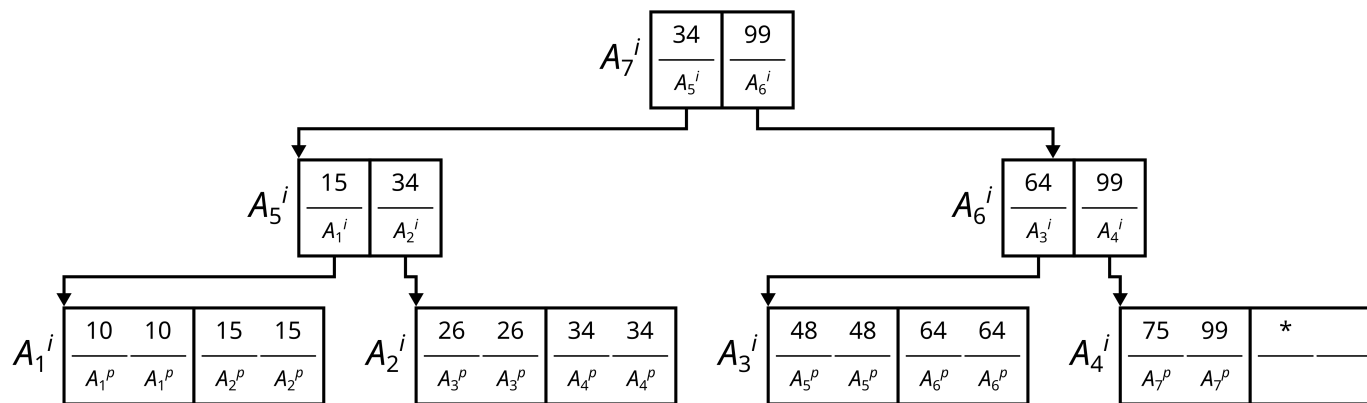
Структура

- Зона прекорачења ^(Mogin, 2008)
 - повезивање прекорачилаца
 - **директно повезивање**
 - листа прекорачилаца је директно повезана с листом стабла тражења, путем показивача у одговарајућем листу стабла тражења
 - стабло тражења је најчешће с преносом највећих вредности
 - информације у вези с прекорачиоцима су у елементима листова стабла тражења
 - елемент листа је четворка (k_e, A_e, k_z, A_z)
 - k_e – тренутно највећа вредност кључа у блоку на адреси A_e
 - A_e – адреса блока примарне зоне
 - k_z – највећа вредност кључа међу прекорачиоцима од блока на адреси A_e
 - A_z – адреса локације оног прекорачиоца од блока на адреси A_e који међу прекорачиоцима од тог блока има најмању вредност кључа

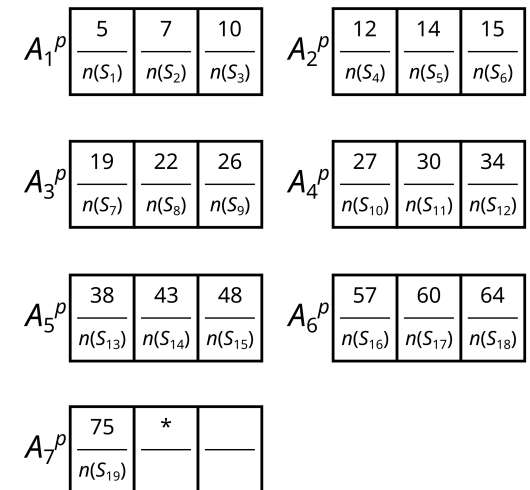
Структура

- Зона прекорачења
 - повезивање прекорачилаца
 - директно повезивање – пример
 - почетно стање

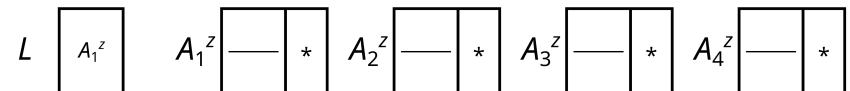
ЗОНА ИНДЕКСА



ПРИМАРНА ЗОНА



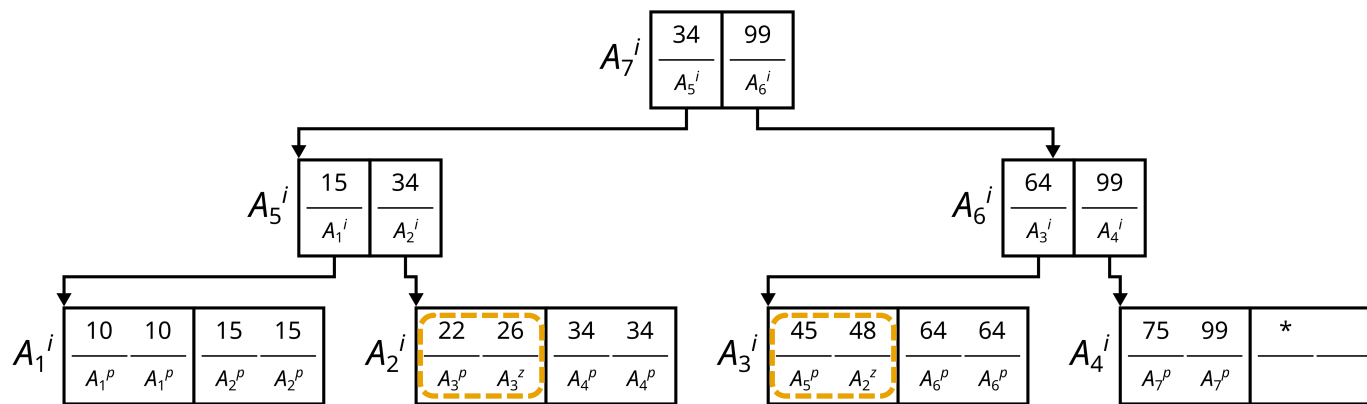
ЗОНА ПРЕКОРАЧЕЊА



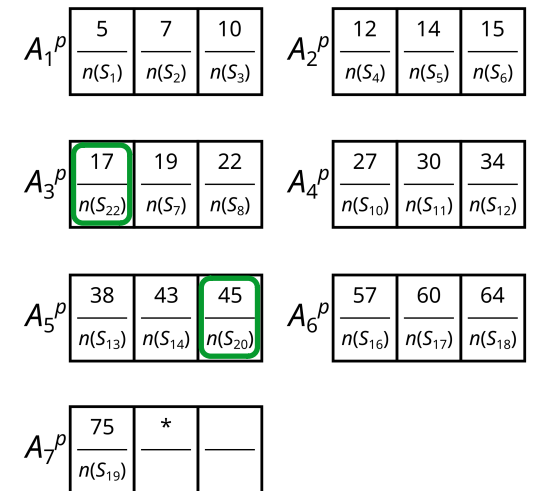
Структура

- Зона прекорачења
 - повезивање прекорачилаца
 - директно повезивање – пример
 - стање по упису слогова S_{20} , S_{21} и S_{22}

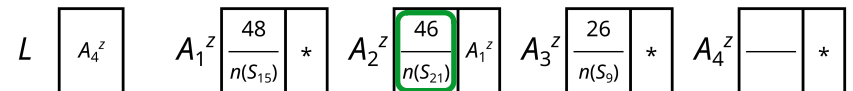
ЗОНА ИНДЕКСА



ПРИМАРНА ЗОНА



ЗОНА ПРЕКОРАЧЕЊА

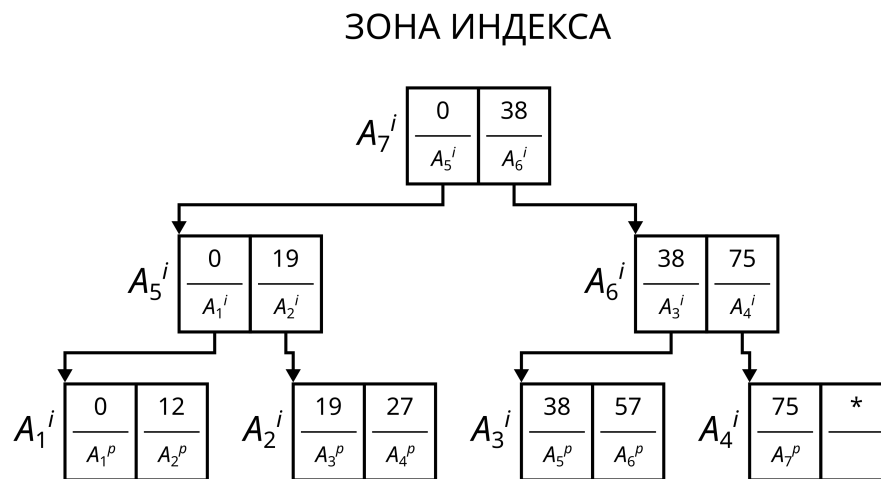


Структура

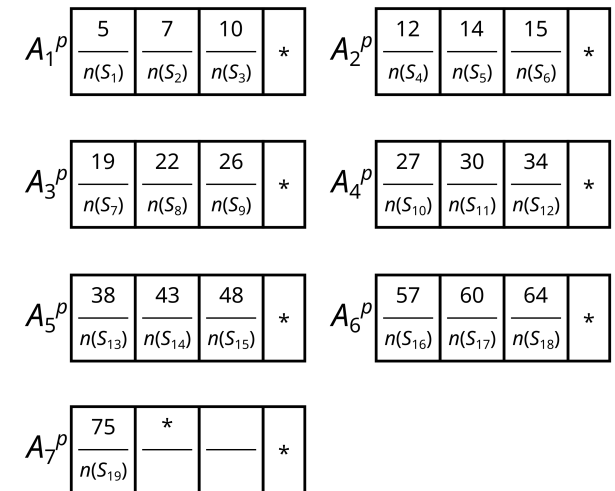
- Зона прекорачења (Mogin, 2008)
 - повезивање прекорачилаца
 - **индиректно повезивање**
 - листа прекорачилаца је индиректно повезана с листом стабла тражења, путем показивача у блоку примарне зоне
 - стабло тражења је најчешће с преносом најмањих вредности
 - информације у вези с прекорачиоцима су у блоковима примарне зоне
 - блок примарне зоне је проширен пољем показивача
 - поље показивача садржи адресу локације оног прекорачиоца од тог блока који међу прекорачиоцима од тог блока има најмању вредност кључа

Структура

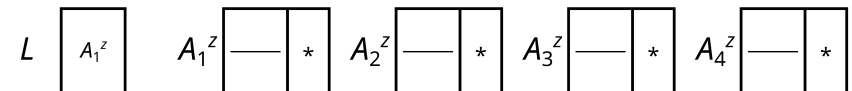
- Зона прекорачења
 - повезивање прекорачилаца
 - индиректно повезивање – пример
 - почетно стање



ПРИМАРНА ЗОНА

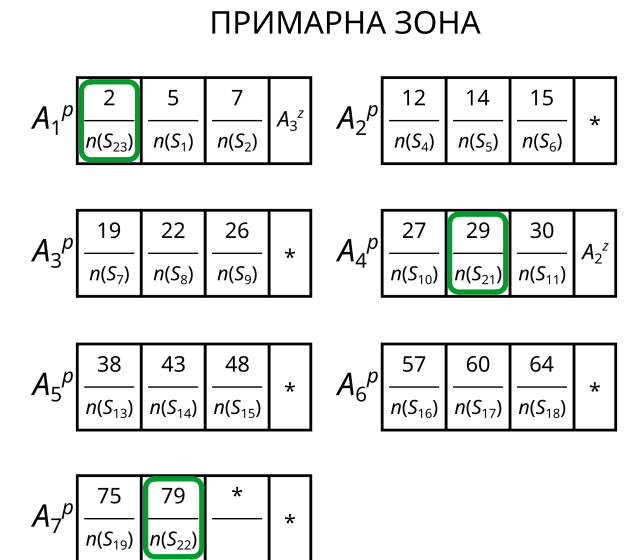
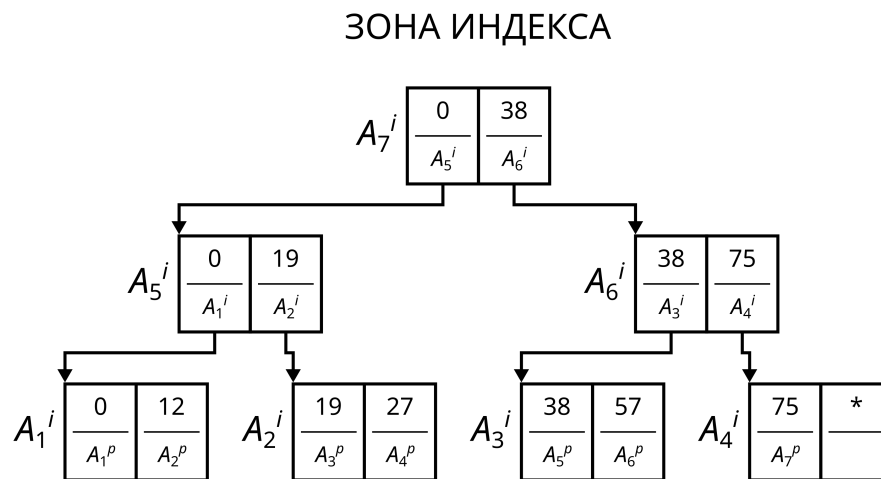


ЗОНА ПРЕКОРАЧЕЊА



Структура

- Зона прекорачења
 - повезивање прекорачилаца
 - индиректно повезивање – пример
 - стање по упису слогова S_{20} , S_{21} , S_{22} , S_{23} и S_{24}



Садржај

- Увод
- Структура
- **Процедуре**
- Одлике и примена
- Ресурси

Процедуре

- **Формирање** (Mogin, 2008)
 - датотеке које су заступљене у поступку формирања
 - серијска датотека D_{ser}
 - секвенцијална датотека D_{sek}
 - индекс-секвенцијална датотека D_{indsek}

Процедуре

- **Формирање** (Mogin, 2008)
 - кораци у поступку формирања
 - 1) на основу датотеке D_{ser} сортирањем у растућем поретку вредности кључа бива формирана датотека D_{sek}

Процедуре

- **Формирање** (Mogin, 2008)
 - кораци у поступку формирања
 - 2) на основу датотеке D_{sek} бива формирана датотека D_{indsek}
 - формирање примарне зоне датотеке D_{indsek}
 - узастопни слогови датотеке D_{sek} бивају читани и на основу њих формиран блокови у примарној зони

Процедуре

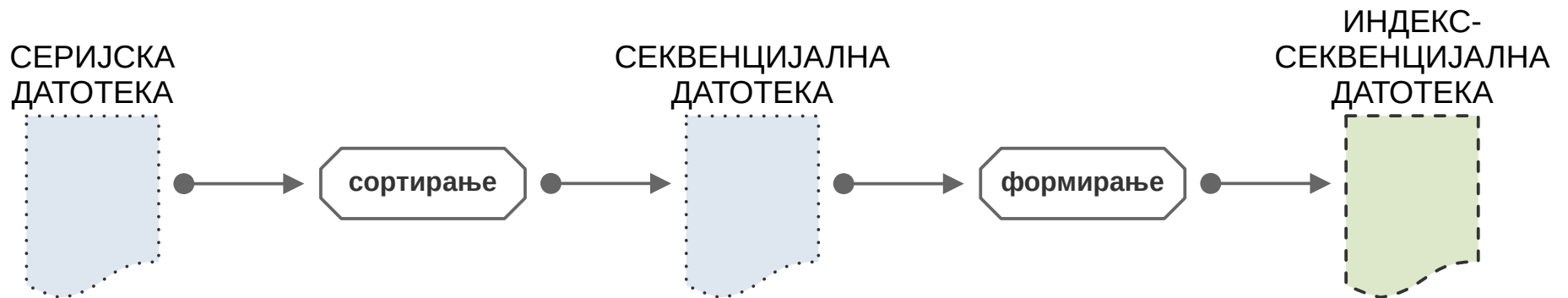
- **Формирање** (Mogin, 2008)
 - кораци у поступку формирања
 - 2) на основу датотеке D_{sek} бива формирана датотека D_{indsek}
 - формирање зоне индекса датотеке D_{indsek}
 - стабло тражења у зони индекса бива формирано од нижих према вишим нивоима, а ниво од левог према десном чвору
 - за сваки формиран блок у примарној зони, нови елемент бива уписан у текући лист зоне индекса, сагласно одабраној врсти преноса вредности
 - по испуњењу једног листа, по потреби припрема се нови
 - по завршетку формирања блокова примарне зоне, долази до формирања виших нивоа у стаблу тражења (када $h > 1$)
 - за групе узастопних чворова на нивоу $i > 1$, формирају се одговарајући директно надређени чворови на нивоу $i - 1$
 - групе узастопних чворова су бројности n , осим последње, која обухвата све преостале чворове нивоа i

Процедуре

- **Формирање** (Mogin, 2008)
 - кораци у поступку формирања
 - 2) на основу датотеке D_{sek} бива формирана датотека D_{indsek}
 - формирање зоне прекорачења датотеке D_{indsek}
 - бива успостављена празна зона прекорачења
 - не долази до уписа слогова у зону прекорачења током формирања

Процедуре

- **Формирање** (Mogin, 2008)
 - ОСНОВНИ ТОК – ПРЕГЛЕД



Процедуре

- Формирање

- пример

- а) серијска датотека

$$A_1 \begin{array}{|c|c|c|} \hline 27 & 38 & 30 \\ \hline n(S_1) & n(S_2) & n(S_3) \\ \hline \end{array}$$
$$A_2 \begin{array}{|c|c|c|} \hline 75 & 14 & 5 \\ \hline n(S_4) & n(S_5) & n(S_6) \\ \hline \end{array}$$
$$A_3 \begin{array}{|c|c|c|} \hline 26 & 19 & 57 \\ \hline n(S_7) & n(S_8) & n(S_9) \\ \hline \end{array}$$
$$A_4 \begin{array}{|c|c|c|} \hline 12 & 64 & 60 \\ \hline n(S_{10}) & n(S_{11}) & n(S_{12}) \\ \hline \end{array}$$
$$A_5 \begin{array}{|c|c|c|} \hline 34 & 10 & 43 \\ \hline n(S_{13}) & n(S_{14}) & n(S_{15}) \\ \hline \end{array}$$
$$A_6 \begin{array}{|c|c|c|} \hline 22 & 48 & 15 \\ \hline n(S_{16}) & n(S_{17}) & n(S_{18}) \\ \hline \end{array}$$
$$A_7 \begin{array}{|c|c|c|} \hline 7 & * & \text{---} \\ \hline n(S_{19}) & \text{---} & \text{---} \\ \hline \end{array}$$

Процедуре

- Формирање

- пример

- б) секвенцијална датотека

A_1	<table border="1"><tr><td>5</td><td>7</td><td>10</td></tr><tr><td>$n(S_1)$</td><td>$n(S_2)$</td><td>$n(S_3)$</td></tr></table>	5	7	10	$n(S_1)$	$n(S_2)$	$n(S_3)$	A_2	<table border="1"><tr><td>12</td><td>14</td><td>15</td></tr><tr><td>$n(S_4)$</td><td>$n(S_5)$</td><td>$n(S_6)$</td></tr></table>	12	14	15	$n(S_4)$	$n(S_5)$	$n(S_6)$
5	7	10													
$n(S_1)$	$n(S_2)$	$n(S_3)$													
12	14	15													
$n(S_4)$	$n(S_5)$	$n(S_6)$													

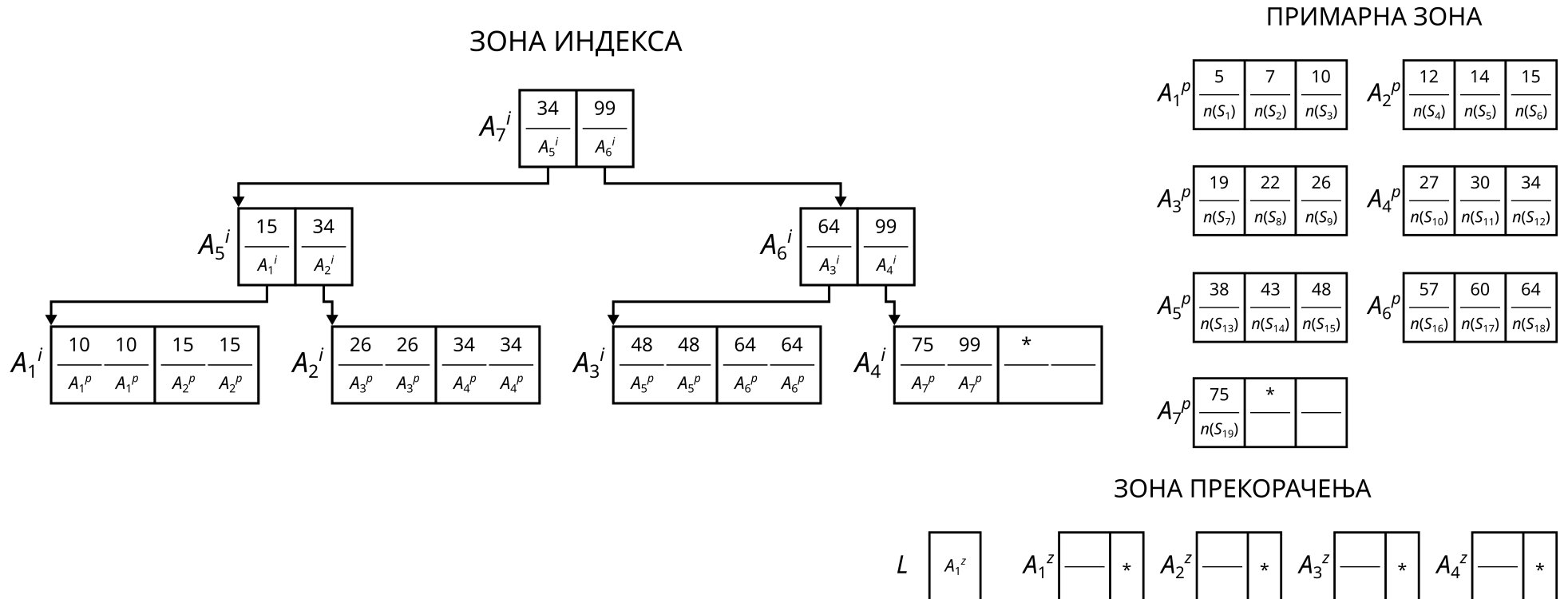
A_3	<table border="1"><tr><td>19</td><td>22</td><td>26</td></tr><tr><td>$n(S_7)$</td><td>$n(S_8)$</td><td>$n(S_9)$</td></tr></table>	19	22	26	$n(S_7)$	$n(S_8)$	$n(S_9)$	A_4	<table border="1"><tr><td>27</td><td>30</td><td>34</td></tr><tr><td>$n(S_{10})$</td><td>$n(S_{11})$</td><td>$n(S_{12})$</td></tr></table>	27	30	34	$n(S_{10})$	$n(S_{11})$	$n(S_{12})$
19	22	26													
$n(S_7)$	$n(S_8)$	$n(S_9)$													
27	30	34													
$n(S_{10})$	$n(S_{11})$	$n(S_{12})$													

A_5	<table border="1"><tr><td>38</td><td>43</td><td>48</td></tr><tr><td>$n(S_{13})$</td><td>$n(S_{14})$</td><td>$n(S_{15})$</td></tr></table>	38	43	48	$n(S_{13})$	$n(S_{14})$	$n(S_{15})$	A_6	<table border="1"><tr><td>57</td><td>60</td><td>64</td></tr><tr><td>$n(S_{16})$</td><td>$n(S_{17})$</td><td>$n(S_{18})$</td></tr></table>	57	60	64	$n(S_{16})$	$n(S_{17})$	$n(S_{18})$
38	43	48													
$n(S_{13})$	$n(S_{14})$	$n(S_{15})$													
57	60	64													
$n(S_{16})$	$n(S_{17})$	$n(S_{18})$													

A_7	<table border="1"><tr><td>75</td><td>*</td><td></td></tr><tr><td>$n(S_{19})$</td><td></td><td></td></tr></table>	75	*		$n(S_{19})$		
75	*						
$n(S_{19})$							

Процедуре

- Формирање
 - пример
 - в) индекс-секвенцијална датотека



Процедуре

- Тражење ^(Mogin, 2008)
 - тражење случајно одабраног слога
 - кораци у поступку тражења
 - 1) тражење адресе у зони индекса
 - за дату вредност кључа k , пролазак кроз стабло тражења једним путем од корена до листа и читавање адресе A у листу
 - адреса блока зоне индекса у којем се налази корен стабла може бити у заглављу датотеке
 - пролазак кроз стабло тражења је заснован на основном поступку тражења адресе у стаблу тражења
 - тражење унутар чвора стабла изводи се по узору на тражење у секвенцијалној датотеци
 - примена линеарне или бинарне методе тражења
 - прочитана адреса у листу стабла представља адресу на којој треба наставити поступак тражења слога

Процедуре

- Тражење ^(Mogin, 2008)
 - тражење случајно одабраног слога
 - кораци у поступку тражења
 - 1) тражење адресе у зони индекса
 - случај директног повезивања
 - примена основног поступка тражења адресе у стаблу тражења уз модификацију за лист
 - елемент листа има структуру (k_e, A_e, k_z, A_z)
 - ако $k \leq k_e$, адреса A је адреса блока примарне зоне A_e
 - ако $k_e < k \leq k_z$, адреса A је адреса блока зоне прекорачења A_z
 - случај индиректног повезивања
 - примена основног поступка тражења адресе у стаблу тражења, чиме у листу стабла тражења бива пронађена адреса блока примарне зоне A

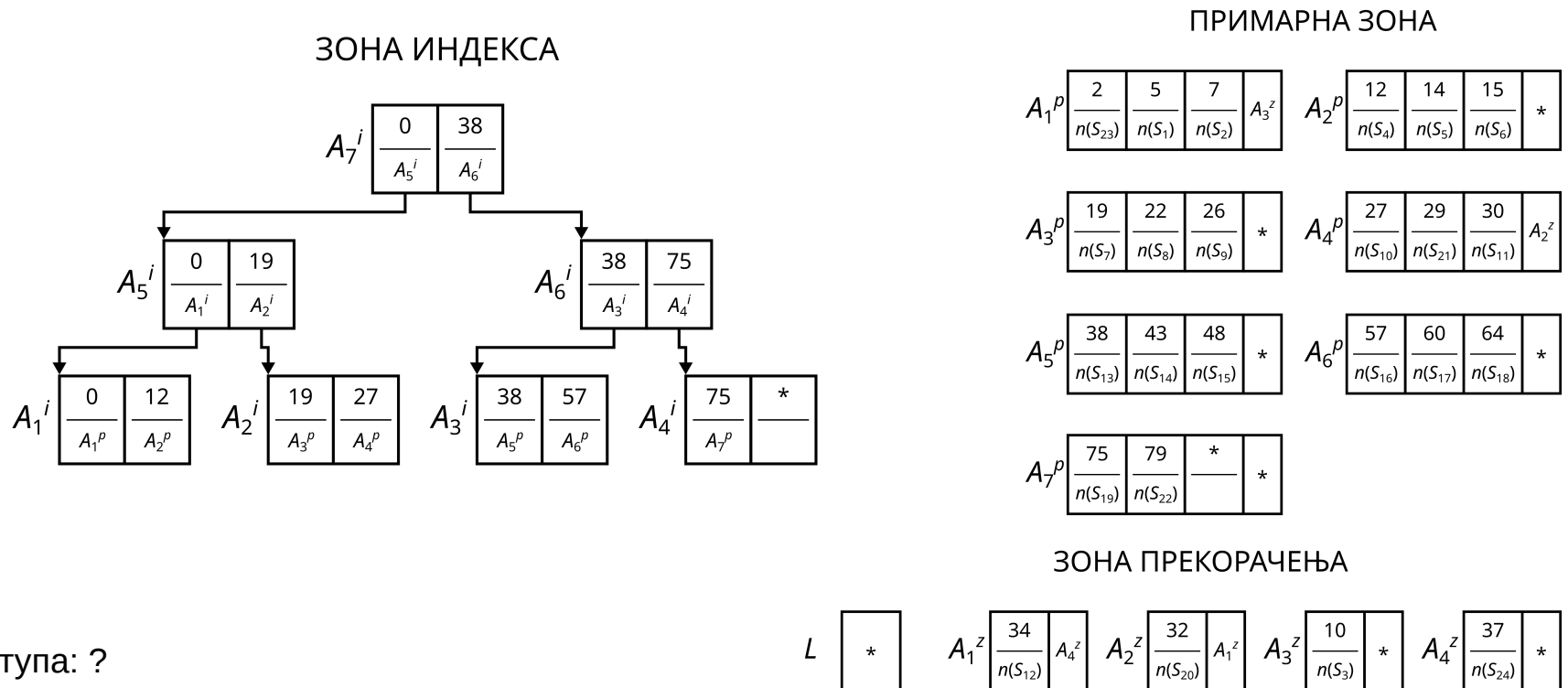
Процедуре

- Тражење ^(Mogin, 2008)
 - тражење случајно одабраног слога
 - кораци у поступку тражења
 - 2) тражење међу слоговима
 - тражење слога одвија се од адресе A добијене у кораку 1
 - случај директног повезивања
 - ако адреса A одговара блоку примарне зоне
 - примена линеарне или бинарне методе тражења у блоку примарне зоне на адреси A
 - ако адреса A одговара блоку зоне прекорачења
 - примена методе праћења показивача од блока на адреси A у зони прекорачења
 - случај индиректног повезивања
 - тражење у блоку примарне зоне на адреси A , а по потреби и у зони прекорачења међу прекорачиоцима од тог блока

Процедуре

- Тражење – пример

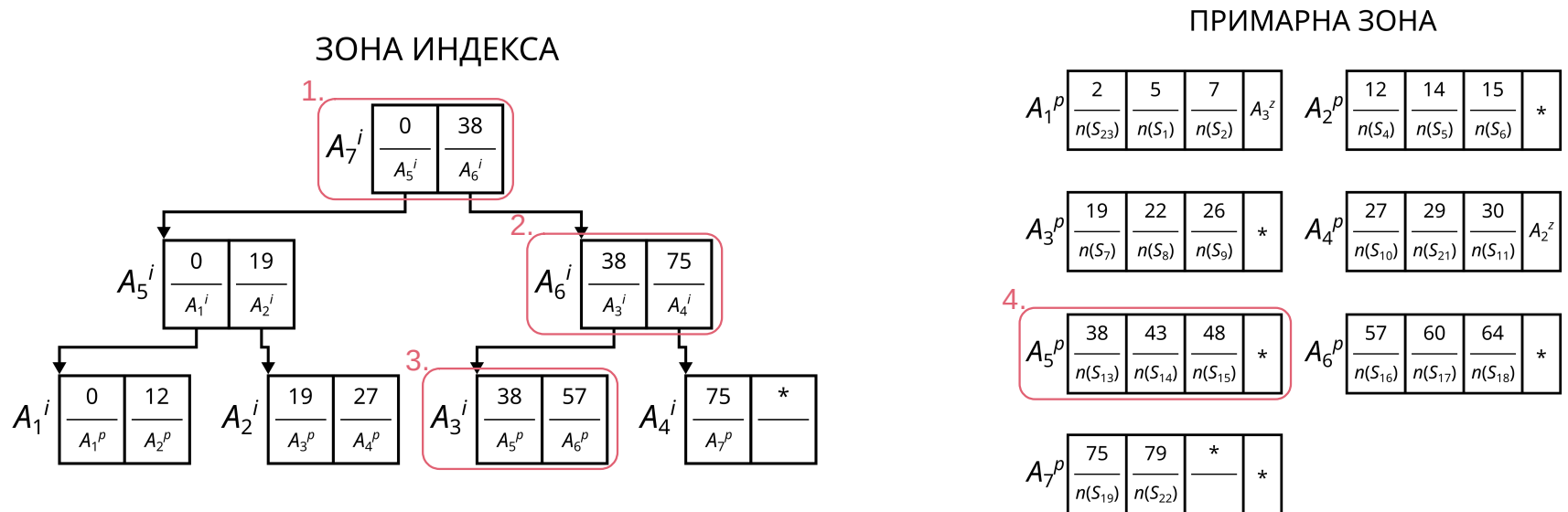
- тражење случајно одабраног слога за вредност кључа $k = 47$
 - (случај индиректног повезивања и преноса најмањих вредности)



Процедуре

- Тражење – пример

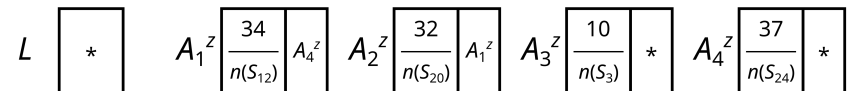
- тражење случајно одабраног слога за вредност кључа $k = 47$
 - (случај индиректног повезивања и преноса најмањих вредности)



исход: неуспешно

број приступа: 4

ЗОНА ПРЕКОРАЧЕЊА



Процедуре

- Тражење ^(Mogin, 2008)
 - тражење логички наредног слога
 - комбинована примена методе линеарног тражења и методе праћења показивача
 - прво тражење почиње од првог блока примарне зоне, а свако наредно тражење наставља се од текућег слога
 - блокови примарне зоне редом бивају проверавани, при чему за сваки блок треба узети у обзир и одговарајућу листу прекорачилаца
 - након проласка преко последњег слога у блоку примарне зоне, даље треба проверити листу прекорачилаца од тог блока, па тек онда може по потреби бити провераван остатак блокова и прекорачилаца

Процедуре

- Тражење ^(Mogin, 2008)
 - тражење логички наредног слога
 - комбинована примена методе линеарног тражења и методе праћења показивача
 - метода линеарног тражења користи се за тражење унутар блока примарне зоне
 - метода праћења показивача користи се за тражење унутар листе прекорачилаца
 - локација показивача на почетак листе прекорачилаца зависи од одабраног поступка повезивања
 - директно повезивање – показивач у листу стабла тражења
 - адреса крајњег левог листа може бити у заглављу датотеке
 - логички суседни листови могу бити у суседним блоковима зоне индекса или међусобно спрегнути
 - индиректно повезивање – показивач у блоку примарне зоне

Процедуре

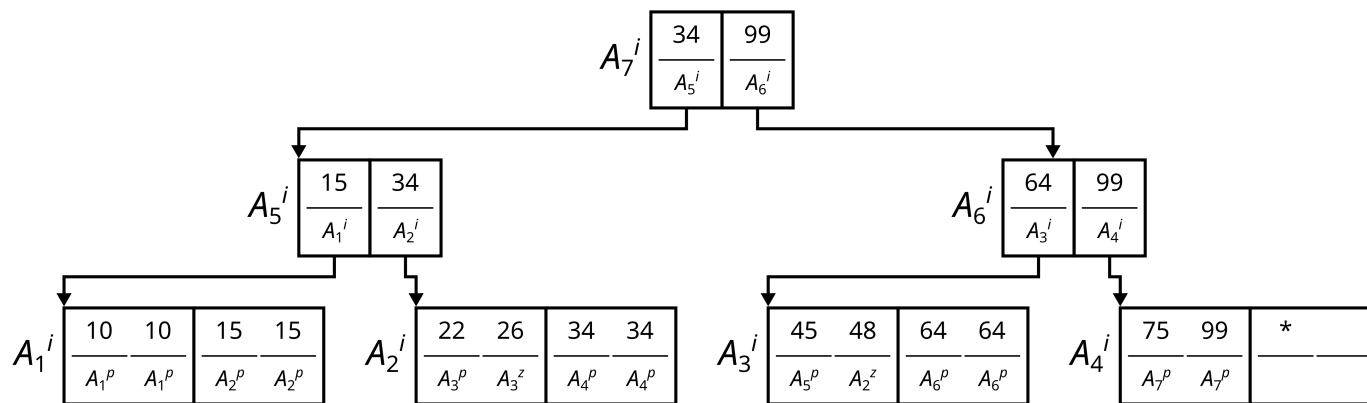
- Тражење – пример

- тражење логички наредног слога за вредност кључа $k = 48$

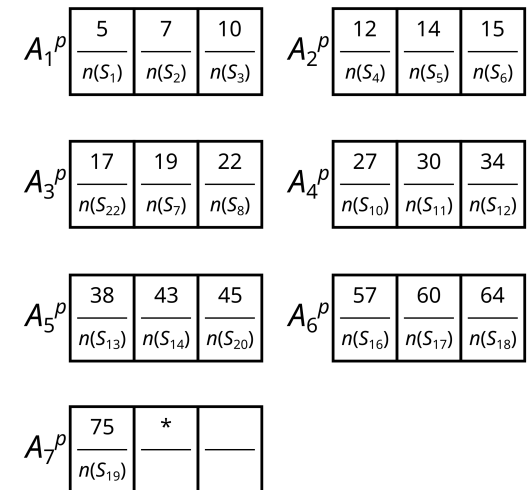
- (случај директног повезивања и преноса највећих вредности)

текући слог је слог S_{11}

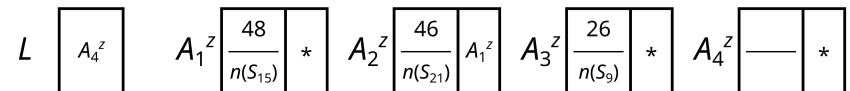
ЗОНА ИНДЕКСА



ПРИМАРНА ЗОНА



ЗОНА ПРЕКОРАЧЕЊА



исход: ?

број приступа: ?

Процедуре

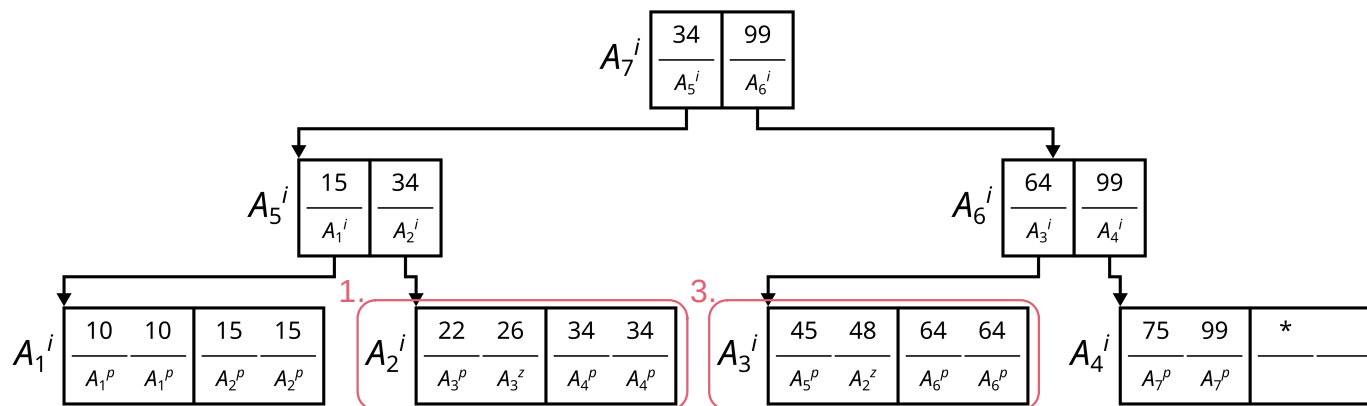
- Тражење – пример

- тражење логички наредног слога за вредност кључа $k = 48$

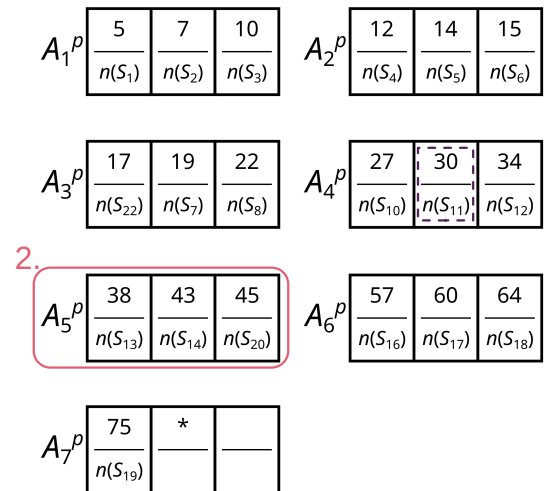
- (случај директног повезивања и преноса највећих вредности)

текући слог је слог S_{11}

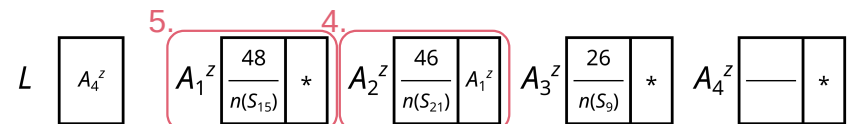
ЗОНА ИНДЕКСА



ПРИМАРНА ЗОНА



ЗОНА ПРЕКОРАЧЕЊА



исход: успешно

број приступа: 5

Процедуре

- Тражење – перформансе (Mogin, 2008)
 - број приступа за тражење случајно одабраног слога R (и за успешно и за неуспешно тражење)
 - случај директног повезивања
$$h+1 \leq R \leq h+z$$
 - z – број прекорачилаца од једног блока
 - случај индиректног повезивања
$$h+1 \leq R \leq h+1+z$$

Процедуре

- Тражење – перформансе (Mogin, 2008)
 - тражење случајно одабраног слога
 - тражење адресе у стаблу тражења у зони индекса захтева h приступа
 - потребно је проћи пут од корена до одговарајућег листа у стаблу тражења и са сваког нивоа учитати по један чвор
 - ако су два бафера на располагању, број приступа зони индекса може бити смањен на $h - 1$
 - корен стабла може бити стално смештен у једном од бафера

Процедуре

- Тражење – перформансе (Mogin, 2008)
 - тражење случајно одабраног слога
 - већа ефикасност у случају директног повезивања него у случају индиректног повезивања
 - у случају индиректног повезивања потребно је приступање зони индекса, након чега долази до приступања примарној зони па евентуално и зони прекорачења
 - у случају директног повезивања потребно је приступање зони индекса, након чега долази до приступања или примарној зони или зони прекорачења
 - на основу садржаја листа стабла тражења у зони индекса, бива одлучено да ли тражење треба да буде настављено у блоку примарне зоне или међу прекорачиоцима у зони прекорачења

Процедуре

- Тражење – перформансе (Mogin, 2008)

- број приступа за тражење логички наредног слога R (и за успешно и за неуспешно тражење)

- случај директног повезивања

$$0 \leq R \leq B + \left\lceil \frac{B}{n} \right\rceil + Z - (i + j + k)$$

- Z – укупни број слогова у зони прекорачења

- i – редни број текућег блока у односу на почетак примарне зоне

- j – редни број текућег листа стабла тражења

$$j = \left\lceil \frac{i}{n} \right\rceil$$

- k – број слогова зоне прекорачења којима је већ приступљено

- случај индиректног повезивања

$$0 \leq R \leq B + Z - (i + k)$$

Процедуре

- Тражење – перформансе (Mogin, 2008)
 - тражење логички наредног слога
 - већа ефикасност у случају индиректног повезивања него у случају директног повезивања
 - у случају директног повезивања може потребно приступање и блоковима примарне зоне и листовима стабла тражења у зони индекса и прекорачиоцима у зони прекорачења
 - у случају индиректног повезивања може потребно приступање и блоковима примарне зоне и прекорачиоцима у зони прекорачења

Процедуре

- Обрада ^(Mogin, 2008)
 - статичка индекс-секвенцијална датотека у улози водеће датотеке
 - погодност за употребу у улози водеће датотеке у режиму редоследне обраде
 - погодност за употребу у улози водеће датотеке у режиму директне обраде

Процедуре

- Обрада ^(Mogin, 2008)
 - статичка индекс-секвенцијална датотека у улози обрађиване датотеке
 - могућност ефикасне употребе у улози обрађиване датотеке у режиму редоследне обраде
 - наизменично приступање блоковима примарне зоне и листама прекорачилаца
 - у случају директног повезивања очекивано је приступање и листовима стабла тражења у зони индекса
 - адреса првог блока примарне зоне може бити у заглављу датотеке
 - могућност ефикасне употребе у улози обрађиване датотеке у режиму директне обраде

Процедуре

- Обрада – перформансе (Mogin, 2008)
 - статичка индекс-секвенцијална датотека у улози обрађиване датотеке
 - редоследна обрада статичке индекс-секвенцијалне датотеке
 - укупни број приступа обрађиваној статичкој индекс-секвенцијалној датотеци R_{uk}
 - ако водећа датотека садржи највећу вредност кључа обрађиване датотеке
 - случај директног повезивања
$$R_{uk} = B + Z + \left\lceil \frac{B}{n} \right\rceil$$
 - Z – укупни број прекорачилаца у обрађиваној датотеци
 - случај индиректног повезивања
$$R_{uk} = B + Z$$

Процедуре

- Обрада – перформансе (Mogin, 2008)
 - статичка индекс-секвенцијална датотека у улози обрађиване датотеке
 - редоследна обрада статичке индекс-секвенцијалне датотеке
 - већа ефикасност у случају индиректног повезивања него у случају директног повезивања
 - разлика у ефикасности је незнатна при уобичајеним вредностима за ред стабла n
 - тражење логички наредног слога је ефикасније у случају индиректног повезивања

Процедуре

- Обрада – перформансе (Mogin, 2008)
 - статичка индекс-секвенцијална датотека у улози обрађиване датотеке
 - директна обрада статичке индекс-секвенцијалне датотеке
 - већа ефикасност у случају директног повезивања него у случају индиректног повезивања
 - разлика у ефикасности није велика
 - тражење случајно одабраног слога је ефикасније у случају директног повезивања

Процедуре

- Ажурирање (Mogin, 2008)
 - ажурирање се изводи у режиму директне обраде
 - примена стабла тражења у проналажењу одговарајуће адресе

Процедуре

- Ажурирање (Mogin, 2008)
 - упис
 - подразумева неуспешно тражење
 - два карактеристична случаја уписа слога
 - **1) нови слог треба сместити у примарну зону**
 - неуспешно тражење се зауставило у блоку примарне зоне без разматрања листе прекорачилаца
 - **2) нови слог треба сместити у зону прекорачења**
 - неуспешно тражење се зауставило у блоку примарне зоне уз разматрање листе прекорачилаца или се зауставило у зони прекорачења

Процедуре

- Ажурирање (Mogin, 2008)
 - упис
 - 1) нови слог треба сместити у примарну зону
 - секвенца слогова која почиње од локације заустављања тражења бива померена за једну локацију надесно унутар блока
 - слогови с вредношћу кључа већом од оне у новом слогу
 - нови слог бива уписан у локацију заустављања тражења
 - ако је пре покушаја уписа блок био комплетан, слог који је до тада имао највећу вредност кључа у блоку треба да буде прекорачилац и да буде премештен у прву слободну локацију зоне прекорачења
 - адреса прве слободне локације је у индексу слободних локација, а по смештању новог прекорачиоца садржај тог индекса очекивано бива увећан за 1
 - нови прекорачилац треба да буде увезан у листу прекорачилаца од тог блока тако да логички поредак слогова буде очуван

Процедуре

- Ажурирање (Mogin, 2008)
 - упис
 - 2) нови слог треба сместити у зону прекорачења
 - нови слог треба да буде уписан као прекорачилац у прву слободну локацију зоне прекорачења
 - адреса прве слободне локације је у индексу слободних локација, а по смештању новог прекорачиоца садржај тог индекса очекивано бива увећан за 1
 - нови прекорачилац треба да буде увезан у листу прекорачилаца од тог блока тако да логички поредак слогова буде очуван

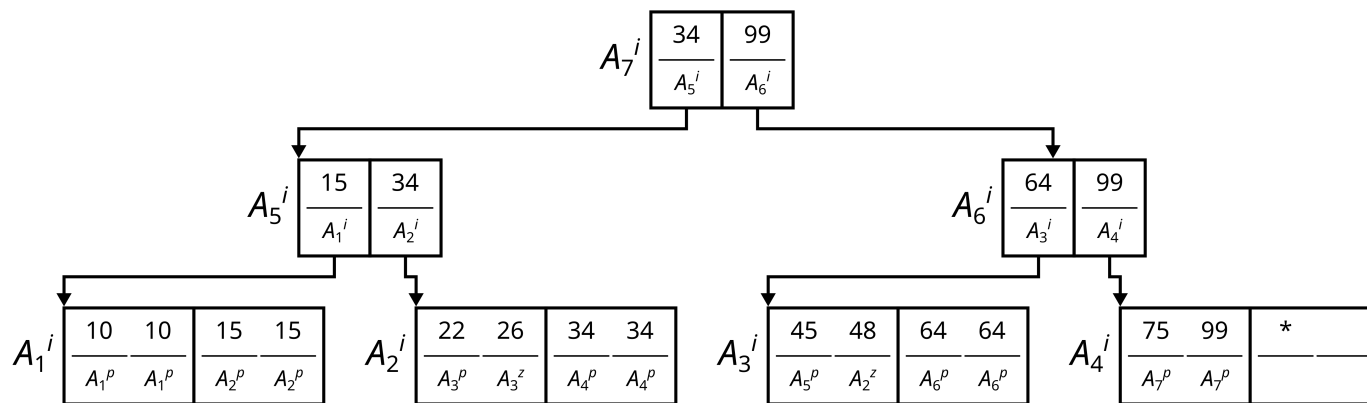
Процедуре

- Ажурирање (Mogin, 2008)
 - упис
 - различити поступци повезивања прекорачилаца са стаблом тражења подразумевају различите посебности о којима треба водити рачуна при упису слога
 - случај директног повезивања
 - треба одржавати актуелним садржај одговарајућег елемента у листу стабла тражења у зони индекса
 - случај индиректног повезивања
 - треба одржавати актуелним садржај поља показивача у одговарајућем блоку примарне зоне

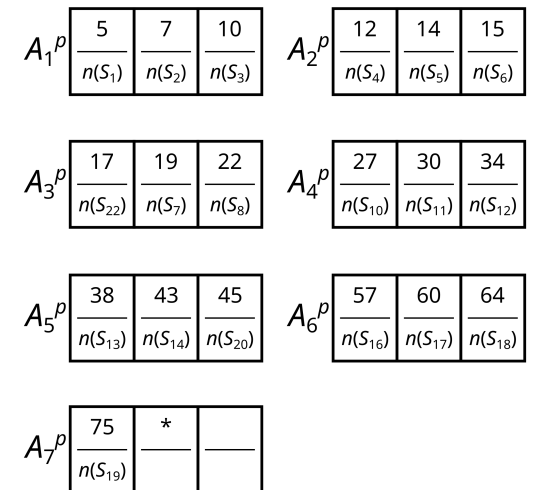
Процедуре

- Ажурирање
 - упис – пример 1
 - стање пре уписа слога с вредношћу кључа $k = 44$

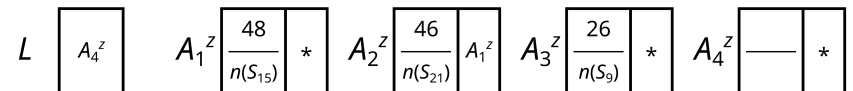
ЗОНА ИНДЕКСА



ПРИМАРНА ЗОНА



ЗОНА ПРЕКОРАЧЕЊА



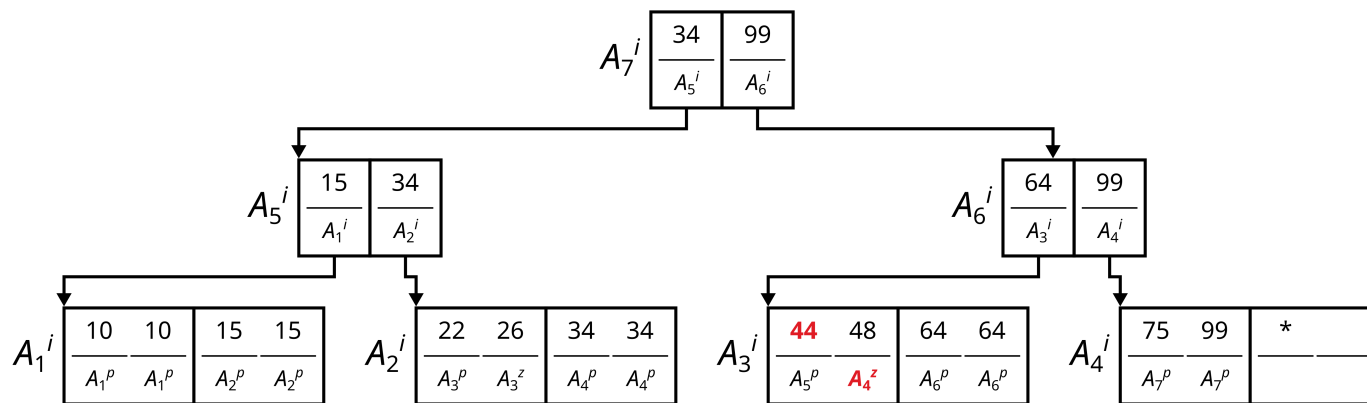
Процедуре

- Ажурирање

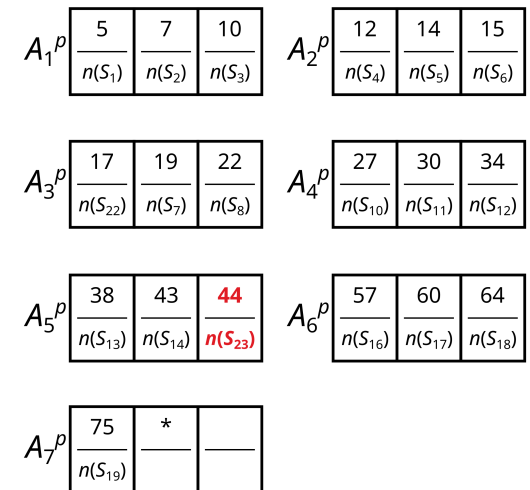
- упис – пример 1

- стање по упису слога с вредношћу кључа $k = 44$

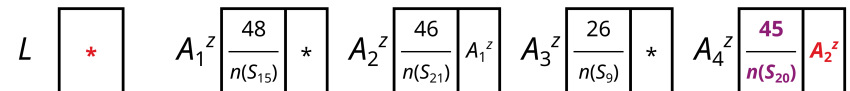
ЗОНА ИНДЕКСА



ПРИМАРНА ЗОНА



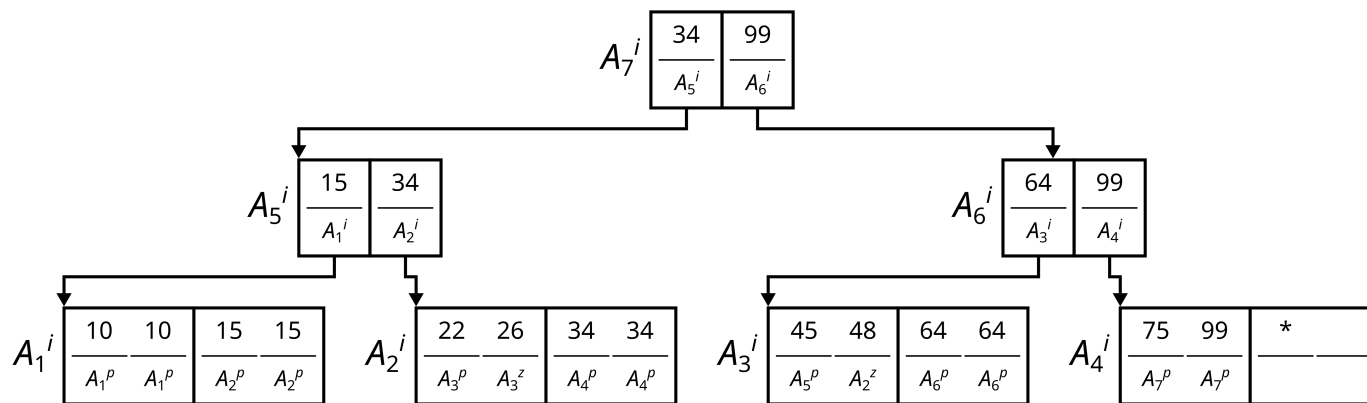
ЗОНА ПРЕКОРАЧЕЊА



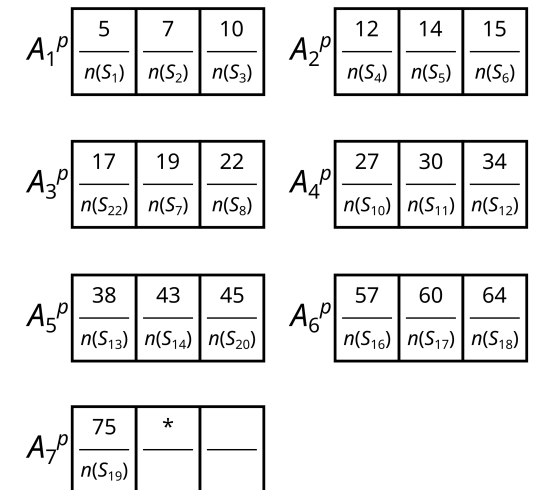
Процедуре

- Ажурирање
 - упис – пример 2
 - стање пре уписа слога с вредношћу кључа $k = 47$

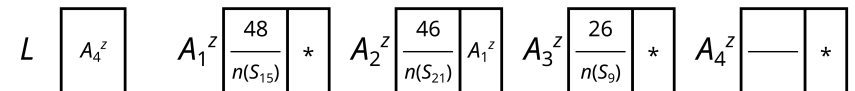
ЗОНА ИНДЕКСА



ПРИМАРНА ЗОНА



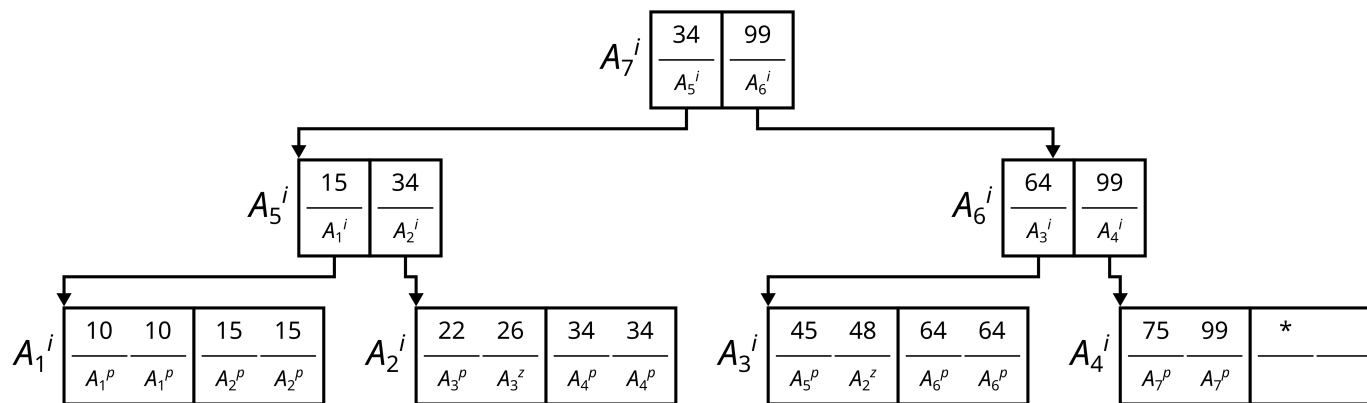
ЗОНА ПРЕКОРАЧЕЊА



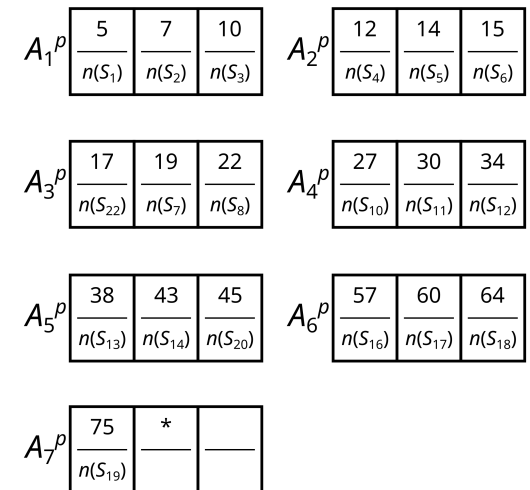
Процедуре

- Ажурирање
 - упис – пример 2
 - стање по упису слога с вредношћу кључа $k = 47$

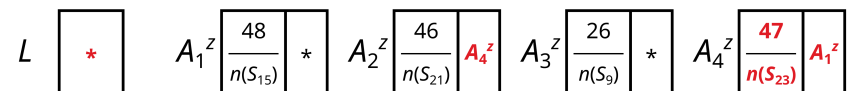
ЗОНА ИНДЕКСА



ПРИМАРНА ЗОНА



ЗОНА ПРЕКОРАЧЕЊА

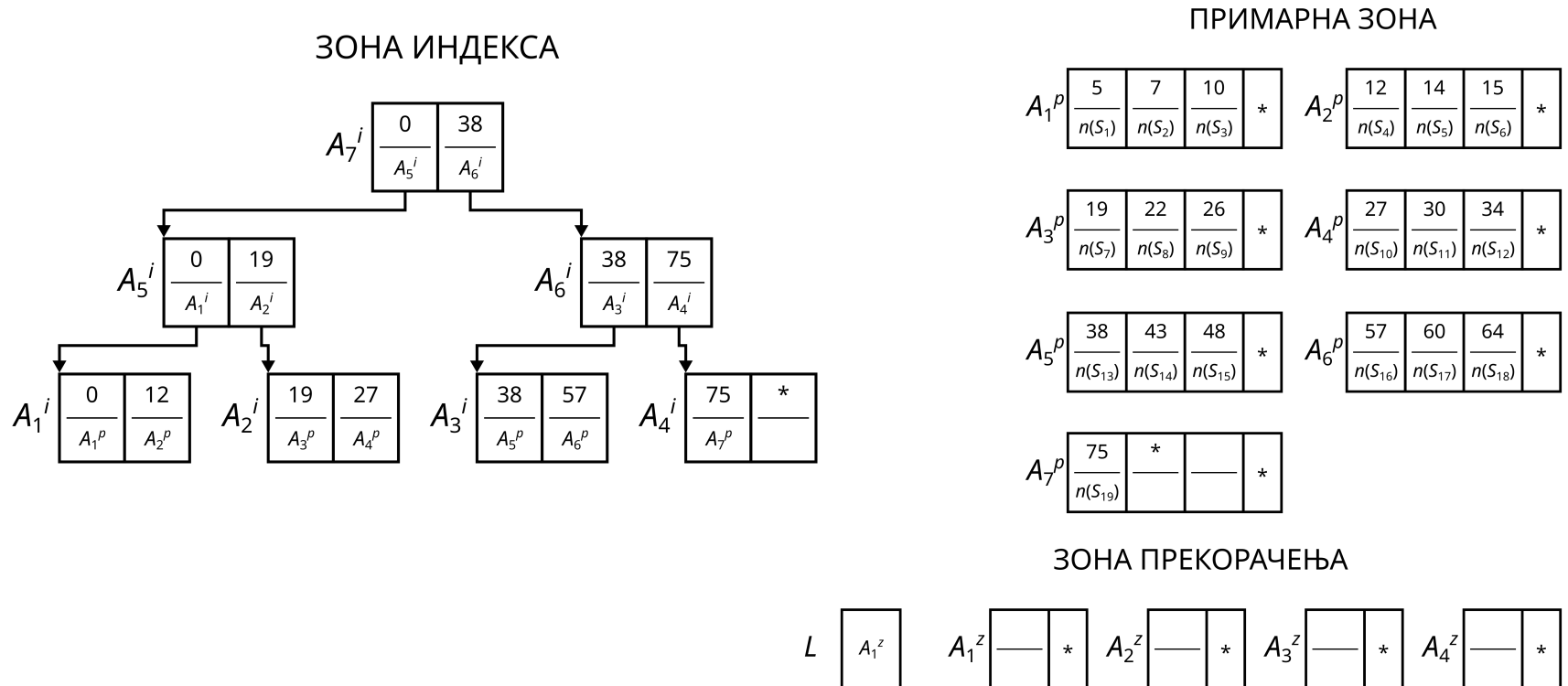


Процедуре

- Ажурирање

- упис – пример 3

- стање пре уписа слога с вредношћу кључа $k = 9$

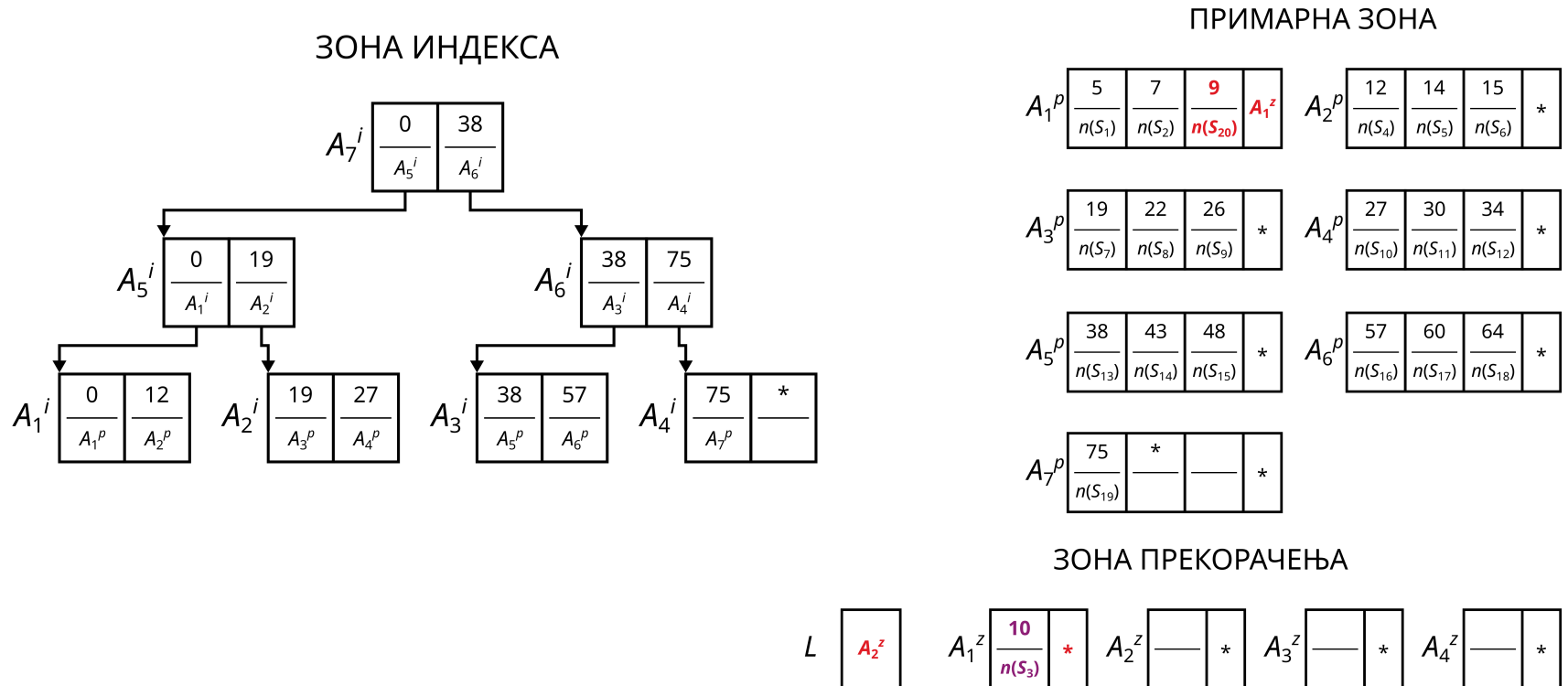


Процедуре

- Ажурирање

- упис – пример 3

- стање по упису слога с вредношћу кључа $k = 9$



Процедуре

- Ажурирање (Mogin, 2008)
 - модификација
 - као у случају модификације за серијску или секвенцијалну организацију, али уз ослањање на тражење које се користи у статичкој индекс-секвенцијалној датотеци
 - примена тражења случајно одабраног слога

Процедуре

- Ажурирање (Mogin, 2008)
 - брисање
 - подразумева успешно тражење
 - најчешће се изводи логичко брисање, због перформанси
 - физичко брисање би могло захтевати померање слогова и ажурирање листе прекорачилаца, што би очекивано подразумевало већи број приступа

Процедуре

- Ажурирање (Mogin, 2008)
 - логичко брисање
 - потребно да структура слога буде проширена пољем статуса
 - вредност поља статуса указује на то да ли је слог актуелан или није
 - након успешног тражења, верзија слога за брисање с измењеном вредношћу статуса бива уписана у постојећу локацију тог слога

Процедуре

- Ажурирање (Mogin, 2008)
 - логичко брисање
 - могућност додатне употребе локације логички обрисаног слога
 - локација логички обрисаног слога може послужити за упис новог слога ако је вредност кључа новог слога у одређеном опсегу
 - ако је локација у блоку зоне података
 - вредност кључа новог слога мора бити мања од највеће вредности кључа у блоку и таква да слог логички припада блоку
 - ако је локација у зони прекорачења
 - вредност кључа новог слога мора бити већа од вредности кључа директног претходника логички обрисаног слога и мања од вредности кључа директног следбеника логички обрисаног слога

Процедуре

- Ажурирање – перформансе (Mogin, 2008)
 - модификација
 - број приступа за успешну модификацију R_m
 $R_m = R_u + 1$
 - логичко брисање
 - број приступа за успешно логичко брисање R_d^l
 $R_d^l = R_u + 1$

Процедуре

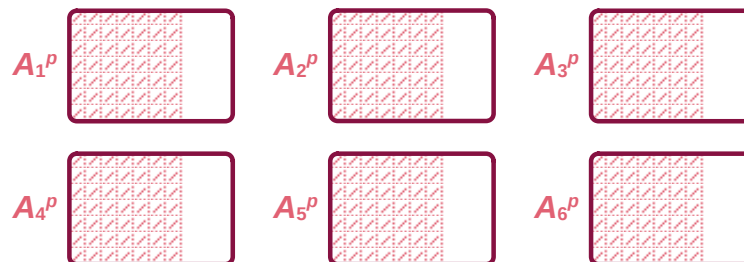
- Реорганизација (Mogin, 2008)
 - потреба за реорганизацијом
 - долази до погоршања перформанси тражења, па и обраде, због уписа слогова у зону прекорачења и логичког брисања слогова
 - већи интензитет ажурирања очекивано води у веће погоршање перформанси
 - ток поступка реорганизације
 - 1) сортирање актуелних слогова из примарне зоне и зоне прекорачења
 - 2) смештање сортираних слогова према успостављеном поретку у нову примарну зону и генерисање новог стабла тражења

Процедуре

- Реорганизација (Mogin, 2008)
 - учесталост реорганизације
 - временски интервал између две реорганизације може бити фиксан
 - нпр. нова реорганизација 30 дана након претходне
 - временски интервал између две реорганизација може варирати зависно од степена попуњености зоне прекорачења
 - нпр. нова реорганизација када зоне прекорачења постане попуњена 75%

Процедуре

- Реорганизација (Mogin, 2008)
 - могућност ублажавања погоршавања перформанси због уписа слогова
 - поступак дистрибуирања слободног простора
 - при формирању датотеке, блокови примарне зоне бивају попуњавани делимично
 - нпр. само 70% капацитета блока бива искоришћено
 - постојање почетног слободног простора у блоковима примарне зоне може одложити појаву прекорачилаца, али не подразумева апсолутно избегавање могућности појаве прекорачилаца
 - меморијски простор није економично искоришћен



Садржај

- Увод
- Структура
- Процедуре
- **Одлике и примена**
- Ресурси

Одлике и примена

- Одлике и примена (Mogin, 2008)
 - предности
 - иницијално, перформансе за улогу обрађиване датотеке у режиму редоследне обраде практично не заостају за оним код секвенцијалних датотека
 - иницијално, перформансе за улогу обрађиване датотеке у режиму директне обраде не заостају значајније за оним код расутих датотека

Одлике и примена

- Одлике и примена (Mogin, 2008)
 - мане
 - услед појаве прекорачилаца долази до погоршања перформанси обраде
 - перформансе редоследне обраде могу бити под контролом ако је зона прекорачења димензионисана на одговарајући начин
 - перформансе директне обраде могу бити посебно погоршане
 - с једним блоком примарне зоне може бити у вези веома много прекорачилаца, што може довести до великог броја приступа у појединим случајевима
 - решење може бити повремено извођење реорганизације, што је неповољно ако је датотека велика

Одлике и примена

- Одлике и примена (Mogin, 2008)
 - примена
 - посебна погодност за случај када исте податке треба обрађивати у различитим режимима обраде, некад директне а некад редоследне
 - интензивна употреба у пакетној обради
 - честа употреба у интерактивној даљинској обради
 - примена физичких структура података заснованих на принципима изградње статичких индекс-секвенцијалних датотека
 - интензивно у мрежним системима за управљање базама података
 - ретко у релационим и објектно-релационим системима за управљање базама података
 - углавном историјски значај

Садржај

- Увод
- Структура
- Процедуре
- Одлике и примена
- **Ресурси**

Ресурси

- Извори и литература
 - Pavle Mogin. Strukture podataka i organizacija datoteka. 3. izdanje. Računarski fakultet (Beograd, Srbija), CET (Beograd, Srbija). 2008.
 - Glava 12. Statička indeks-sekvencijalna organizacija datoteke