Matematička indukcija - dokazivanje korektnosti algoritma

1. Neka je P funkcija koja prirodni broj n preslikava u prirodni broj sa istim ciframa, ali u obrnutom poretku (npr: $P(12345) = 54321$). Konstruisati algoritam koji za ulaznu vrednost - prirodan broj n, izračunava vrednost $P(n)$. Dokazati korektnost napisanog algoritma.
(* pseudo PASCAL-like kôd*)

REŠENJE:

```
Algoritam INVERZIJA (n);
input: n;
output: m;
begin
m:=0;
k:=n;
i:=0;

while k>0 do
begin
   m:=m*10+k mod 10;
   k:=k div 10;
    i:=i+1;
end

end.
```

Dokaz izvodimo za brojeve n koji se ne završavaju nulom.

Invarijanta petlje je relacija izmedju promenljivih koja važi nakon svakog izvršenja bloka naredbi u okviru petlje.

Dokažimo primenom matematičke indukcije da je relacija $n = k \cdot 10^i + P(m)$ invarijanta petlje, tj. dokažimo da ova relacija važi pre ulaska u petlju i nakon svakog izvršenja bloka naredbi u okviru petlje. Takođe dokažimo i da je iskaz *"i je broj cifara broja m"* invarijanta petlje.

Baza: Za $i = 0$ (dakle pre prvog izvršenja petlje) imamo:

$$n = k \cdot 10^0 + P(m);$$
$$n = k + P(0);$$
$$n = n + 0\ ,$$

i m ima nula cifara (smatramo da broj 0 ima nula cifara) pa tvrđenje važi.

IH: Pretpostavimo da tvrđenje važi za i (tj. da važi nakon i izvršenja bloka naredbi u okviru petlje) i dokažimo da važi i za $i + 1$. Dakle, na osnovu pretpostavke važi $n = k \cdot 10^i + P(m)$.
Naredna vrednost promenljive k je k div 10, pa treba dokazati
$n = k' \cdot 10^{i'} + P(m')\ ,\ \Leftrightarrow$
$n = (k \text{ div } 10) \cdot 10^{i+1} + P(m \cdot 10 + (k \bmod 10))\ ,\ \Leftrightarrow$
$n = (k \text{ div } 10) \cdot 10^{i+1} + P(m) + (k \bmod 10) \cdot 10^i\ ,\ \Leftrightarrow$
$n = ((k \text{ div } 10) \cdot 10 + (k \bmod 10)) \cdot 10^i + P(m)\ ,\ \Leftrightarrow$
$n = k \cdot 10^i + P(m)$
što je tačno na osnovu induktivne pretpostavke.
Pri prelasku na treći red koristili smo induktivnu pretpostavku da je i broj cifara broja m.
Naravno, i ovo tvrđenje treba dokazati za nove vrednosti m' i i'.
Tvrđenje važi, jer je $i' = i + 1$, $m' = m \cdot 10 + k \bmod 10$.

Dokažimo još i da algoritam završava rad. Na početku je $k > 0$ i nakon svakog prolaska kroz petlju k se smanjuje, tako da će u konačnom broju koraka k dostići vrednost 0.
Tada algoritam završava sa radom i za $k = 0$ imamo da je $n = P(m)$, pa je $P(n) = m$
(jer za svako $x \in N$, $P(P(x)) = x$), gde je $P(m)$ funkcija koja prirodan broj n preslikava u prirodan broj sa istim ciframa, ali u obrnutom poretku).

Napomena: Ako bismo želeli da obuhvatimo i brojeve koji se završavaju jednom ili sa više nula, algoritam bi bio isti, ali bi dokaz morao da se izmeni. Umesto pomenute invarijante petlje, trebalo bi uzeti relaciju

$n = k \cdot 10^i + P(m) \cdot 10^l$, gde je l broj nula kojima se završava broj n.

2. Konstruisati algoritam koji konvertuje dekadni broj u njegov oktalni zapis. Dokazati korektnost tog algoritma.

```
 Algoritam Okt_cifre(n);
   Input: n
   Output: b (niz oktalnih cifara broja n)
   begin
    t:=n; k:=0;
    while (t>0) do
    begin
      k:=k+1; b[k]:=t mod 8; t:=t div 8;
    end
   end,
```

Invarijanta petlje: Neka je m broj cije su oktalne cifre dobijene nakon k prolazaka kroz while ciklus, pri cemu b[1] je najniza oktalna cifra. Tada je $n=t\cdot 8^k+m$

Baza: za k=0, vazi da t=n, m=0, te je ovo tvrdjenje tacno

Induktivni korak: Neka je ovo tvrdjenje tacno za neko $k \neq 0$

U narednom prolasku kroz ciklus dobijaju se nove vrednosti

b[k+1]=t mod 8, t'=t div 8, $m=b[k+1] \cdot 8^k+m=(t \bmod 8) \cdot 8^k+m$, te je

$t'\cdot 8^{k'}+m' = t'\cdot 8^{k+1}+m'=(t \operatorname{div} 8) \cdot 8^{k+1}+(t \bmod 8) \cdot 8^k+m=8^k(8\cdot(t \operatorname{div} 8) + (t \bmod 8)) +m= t\cdot 8^k+m=$
$=n$ (zbog induktovne hipoteze)

Da li algoritam terminira rad, tj. da li će u nekom trenutku biti t=0?

Niz $t_k=t_{k-1}$ div 8 je monotono opadajući niz prirodnih brojeva, te je po algebarskom principu minimalnog elementa ograničen odozdo nulom.

Kako je posle poslednjeg prolaska kroz petlju t=0, u tom trenutku će biti m=n, čime je korektnost algoritma dokazana.

3. Konstruisati algoritam za konverziju binarnog zapisa nekog broja u oktalni i dokazati korektnost napisanog algoritma.

REŠENJE:

Brojevi u binarnom zapisu su predstavljeni nizom nula i jedinica, a brojevi u oktalnom zapisu nizomcifara od 0 do 7. Za konverziju binarnog zapisa nekog broja *n* u oktalni primenjuje se sledeći algoritam:

```
Algoritam KonvBinOkt
Input: b[],k /* binarni zapis broja n i broj cifara binarnog zapisa */
Output: o[],i /* oktalni zapis broja n, broj cifara */
begin
   d:=0;
   l:=0;

while l<k do
 begin
    d:=d+2^l*b[l+1];
     l:=l+1;
end

t:=d;
i:=0;

while t>0 do
begin
    i=i+1;
    o[i]:=t mod 8;
```

```
    t:=t div 8;
end

end.
```

U prvoj petlji se na osnovu binarnog zapisa $b[]$ ($b[1]$ je cifra najmanje težine) izračunava dekadna vrednost d.
Pre ulaska u petlju i nakon svakog izvršenja prve petlje, među promenljivama važi invarijanta:

vrednost d jednaka je broju čiji je binarni zapis niz $b[1...l]$

Dokaz ovog tvrdjenja izvodimo indukcijom po broju izvršavanja petlje.

Baza: Pre ulaska u petlju promenljive imaju vrednost: $d = 0$, $l = 0$, pa tvrdjenje važi.
IH: Pretpostavimo da je pre nekog izvršenja petlje vrednost d jednaka broju koji odgovara binarnom zapisu $b[1...l]$ i dokažimo da relacija važi i nakon tog izvršenja petlje:
Dekadni broj koji odgovara binarnom zapisu $b[1...l + 1]$ jednak je zbiru broja kojem odgovara binarni zapis $b[1...l]$ i broja $2^l \cdot b[l + 1]$.

Na osnovu induktivne pretpostavke promenljiva d pre ulaska u petlju ima vrednost kojoj odgovara binarni zapis $b[1...l]$, pa nakon izvršenja instrukcije d:=d+2^l*b[l+1] i nakon izvršenja petlje promenljiva d ima vrednost kojoj odgovara binarni zapis $b[1...l + 1]$.
Uslov izlaska iz petlje se postiže, jer će l posle konačnog broja koraka dobiti vrednost k. Nakon izlaska iz petlje vrednost d jednaka je broju čiji je binarni zapis niz $b[1...k]$.

U drugoj petlji se izračunava oktalni zapis $o[]$ dekadnog broja d (pri čemu je $o[1]$ cifra najmanje težine). Pre ulaska u petlji i nakon svakog izvršenja druge petlje medju promenljivama vazi relacija:

$$d = t \cdot 8^i + m$$

gde je m ceo broj sa oktalnom reprezentacijom $o[1...i]$.
Dokaz izvodimo indukcijom po broju izvršavanja druge petlje. Pre prvog izvršavanja druge petlje promenljive imaju vrednost $t = d$, $i = 0$, $m = 0$, pa važi $d = d \cdot 8^0+0$.
Pretpostavimo da pre nekog izvršenja druge petlje važi relacija $d = t \cdot 8^i + m$ i dokažimo da relacija važi i nakon tog izvršenja druge petlje:

$d = (t\ div\ 8) \cdot 8^{i+1} + m + (t\ mod\ 8) \cdot 8^i$

$= \frac{t - (t \bmod 8)}{8} \cdot 8^{i+1} + m + (t\ mod\ 8) \cdot 8^i$

$= t \cdot 8^i - (t\ mod\ 8) \cdot 8^i + m + 8^i \cdot (t\ mod\ 8)$

$= t \cdot 8^i + m$,

a to važi na osnovu induktivne hipoteze.

Uslov izlaska iz petlje se postiže, jer će t posle konačnog broja koraka dobiti vrednost 0. Nakon izlaska iz petlje važi: $t = 0$, $d = 0 \cdot 8^i + m$ tj. $d = m$, a u formulaciji invarijante m je dekadni broj čiju oktalna reprezentaciju sadrži niz $o[]$.

Dakle, nakon izvrešenja algoritma broj čiji je binarni zapis niz $b[]$ jednak je broju čiji je oktalni zapis broj $o[]$, čime je dokazana korektnost algoritma.

(II način:)

```
Algoritam KonvBinOkt2
Input b[], k /* binarni zapis broja n i broj cifara binarnog zapisa (3|k)*/
Output o[],l /* oktalni zapis broja n, l je broj cifara */
begin
l=0;

while 3*l<k do
begin
    o[l+1]:=b[3*l+1]+2*b[3*l+2]+4*b[3*l+3];
     l=l+1;
```

```
end

end.
```

Binarni zapis broja uvek možemo dopuniti nulama, tako da broj cifara bude deljiv sa tri. Zato možemo pretpostaviti da je broj k uvek deljiv sa 3. Neka su $b[1]$ i $o[1]$ cifre najmanje težine nizova $b[]$ i $o[]$.

Pre ulaska u petlju i nakon svakog prolaska kroz petlju među promenljivama važi invarijanta:
broj čiji je oktalni zapis $o[1..l]$ *jednak je broju čiji je binarni zapis* $b[1..3l]$
(l je tekuća promenljiva u petlji).

Dokaz izvodimo indukcijom po broju izvršavanja bloka naredbi u okviru petlje.
Baza: Pre ulaska u petlju je $l = 0$, pa tvrdjenje važi.

IH: Pretpostavimo da je pre nekog izvršenja bloka naredbi vrednost dekadnog broja o čiji je oktalni zapis niz $o[1..l]$ jednak broju čiji je binarni zapis niz $b[1..3l]$ i dokažimo da ta relacija važi i nakon tog izvršenja bloka naredbi u okviru petlje:
Dekadni broj koji odgovara oktalnom zapisu $o[1..l + 1]$ jednak je zbiru broja kojem odgovara oktalni zapis $o[1..l]$ i broja $8^l \cdot o[l + 1]$.
Dekadni broj koji odgovara binarnom zapisu $b[1..3(l + 1)]$ jednak je zbiru broja kojem odgovara binarni zapis $b[1..3l]$ i broja $8^l \cdot (b[3l + 1] + 2b[3l + 2] + 4b[3l + 3])$.

Na osnovu induktivne pretpostavke, broj čiji je oktalni zapis $o[1..l]$ jednak je broju čiji je binarni zapis $b[1..3l]$, a nakon izvršenja komande o[l+1]:=b[3*l+1]+2*b[3*l+2]+4*b[3*l+3]; važi
$8^l \cdot o[l+1] = 8^l \cdot (b[3l+1]+2b[3l+2]+4b[3l+3])$,
pa je broj čiji je oktalni zapis $o[1..l + 1]$ jednak broju čiji je binarni zapis $b[1..3(l + 1)]$,
tj. za novu vrednost l (nakon komande l=l+1;)
važi da je broj čiji je oktalni zapis $o[1..l]$ jednak broju čiji je binarni zapis $b[1..3l]$, što je i trebalo dokazati.

Uslov izlaska iz petlje se postiže jer će $3l$ dobiti vrednost k posle konačnog broja koraka. Nakon izlaska iz petlje u nizu $o[]$ će biti smeštena oktalna reprezentacija broja čiji je binarni zapis niz $b[1..3l]$, tj. $b[1..k]$

4. Napisati algoritam za konverziju oktalnog zapisa nekog broja u heksadecimalni. Dokazati korektnost napisanog algoritma.

5. Napisati algoritam za određivanje najvećeg zajedničkog delioca dva prirodna broja i dokazati korektnost napisanog algoritma.

REŠENJE:

Lema. NZD(a, b) = NZD(b, r), r = a mod b
Dokaz:
$a = q \cdot b + r$, NZD(b, r) | b, r => NZD(b, r) | a
Dalje, NZD(b, r) | a, NZD(b, r) | b => NZD(b, r) | NZD(a, b)

Slicno,
$r = a - q \cdot b$, NZD(a, b) | a, b => NZD(a, b) | r
Dalje, NZD(a, b) | b , NZD(a, b) | r => NZD(a, b) | NZD(b, r)

dakle,
NZD(b, r) | NZD(a, b) , NZD(a, b) | NZD(b, r) => NZD(a, b) = NZD(b, r)
sto je i trebalo dokazati.

Algoritam NZD(m,n);
input: m,n;

```
output: nzd; /* najveci zajednicki delioc brojeva m i n */
begin
   a:=max(m,n);
   b:=min(m,n);
   r:=b;

while r>0 do
begin
   r:=a mod b;
   a:=b;
   b:=r;
end;

nzd:=a;
end.
```

Invarijanta ove petlje je nzd(m, n) = nzd(a, b).
Baza: Pre ulaska u petlju tvrđenje važi, jer je nzd(a, b) = nzd(max(m, n), min(m, n)) = nzd(m, n).
IH:
Pretpostavimo da tvrdjenje važi pre nekog izvršenja bloka naredbi u okviru petlje i dokažimo da važi i nakon tog izvršenja. Dakle, pretpostavimo da pre nekog izvršenja bloka naredbi u okviru petlje važi nzd(m, n) = nzd(a, b).
Nakon izvršenja bloka naredbi u okviru petlje, promenljiva a dobija vrednost a' = b, a promenljiva b vrednost b' = a mod b.

Kako vazi nzd(a', b') = nzd(b, a mod b) = nzd(a, b) i kako je, na osnovu induktivne pretpostavke nzd(m, n) = nzd(a, b), sledi nzd(m, n) = nzd(a', b'), sto je i trebalo dokazati.

Nakon svakog izvrsenja bloka naredbi u okviru petlje promenljiva r ima strogo manju vrednost nego pre njega (jer je r' = a mod b = a mod r < r).
Dakle niz vrednosti promenljive r je strogo opadajuci niz nenegativnih celih brojeva, pa nakon konacnog broja koraka vrednost promenljive r dostici ce vrednost 0 i tada algoritam zavrsava sa radom.

Na osnovu svojstava invarijante, tada vazi nzd(m, n) = nzd(a, b) = nzd(a, 0) = a.
Dakle, nakon izvrsenja petlje promenljiva a ima vrednost nzd(m, n), pa tu vrednost nakon komande nzd:=a ima i promenljiva nzd, sto je i trebalo dokazati.

6. Napisati algoritam koji konvertuje binarni zapis broja u sam taj broj. Dokazati korektnost napisanog algoritma.

7. Algoritam elemente niza različitih brojeva A[1..k] sortira (u rastućem poretku) i rezultat smešta u niz B na sledeći način: u i-tom koraku (1 ≤ i ≤ k) određuje najmanji element niza A, upisuje ga u prvu slobodnu lokaciju u nizu B i taj element briše iz niza A. Konstruisati opisani algoritam i dokazati njegovu korektnost.

```
Algoritam sortiranje(A[1..k],B[1..k])
Input : A (niz), k
Output: B (sortirani niz)

begin
  i:=0;
   while i<k do /* while petlja (1) */
     begin
        i:=i+1;
        m:=1;
        l:=1;
```

```
        while (l<k-i+1) do /* while petlja (2) */
          begin
             l:=l+1;
             if (A[l]<A[m]) then /* Trazimo minimalni element niza A */
             m:=l;
          end;
        B[i]:=A[m];
        A[m]:=A[k-i+1]; /* Izbacivanje najmanjeg elementa niza A */
      end;
end.
```

Invarijanta petlje je relacija izmedju promenljivih koja važi nakon svakog izvršenja bloka naredbi u okviru petlje.
Invarijanta petlje (2) je relacija A[m] = min A[j] , j∈ {1,2,...,l}

Baza: Pre prvog ulaska u petlju to tvrdjenje važi jer je m = 1 i l = 1, pa je A[1] = min A[j], j∈ {1}

IH: Pretpostavimo da tvrdjenje važi pre nekog izvršenja bloka naredni u okviru petlje (tj. pretpostavimo da važi A[m] = min A[j]), j∈ {1,2,...,l} i dokažimo da važi i nakon tog izvršenja bloka naredbi u okviru petlje. Neka su m' i l' vrednosti promenljivih m i l nakon sledeceg izvrsenja bloka naredbi u okviru petlje. Postoje dve mogucnosti:

$A[l'] < A[m]$: tada je $m' = l'$ (i $l' = l + 1$), pa iz induktivne pretpostavke $A[m] = \min_{j\in\{1,2,\ldots,l\}} A[j]$ sledi $A[m'] = A[l'] < A[m] = \min_{j\in\{1,2,\ldots,l'-1\}} A[j]$, odakle je $A[m'] = \min_{j\in\{1,2,\ldots,l'\}} A[j]$, što je i trebalo dokazati.

$\neg(A[l'] < A[m])$: tada je $m' = m$ (i $l' = l + 1$), pa iz induktivne pretpostavke $A[m] = \min_{j\in\{1,2,\ldots,l\}} A[j]$ sledi $A[m'] = A[m] = \min_{j\in\{1,2,\ldots,l'-1\}} A[j]$. Kako je $A[l'] > A[m]$, sledi da je $\min_{j\in\{1,2,\ldots,l'-1\}} A[j] = \min_{j\in\{1,2,\ldots,l'\}} A[j]$, pa je $A[m'] = \min_{j\in\{1,2,\ldots,l'\}} A[j]$, što je i trebalo dokazati.

Dakle, važi $A[m] = \min_{j\in\{1,2,\ldots,l\}} A[j]$ nakon svakog izvršenja bloka naredbi u olviru petlje (2). Nakon svakog izvršenja bloka naredbi vrednost promenljive l povećava se za 1, a vrednosti promenljivih k i i se ne menjaju, pa promenljiva l u konačnom broju koraka dostiže vrednost $k - i + 1$, što znači da petlja staje sa radom nakon konačnog broja koraka. Kada petlja završi rad, promeljiva l ima vrednost $k - i + 1$, pa na osnovu svojstava invarijante sledi $A[m] = \min_{j\in\{1,2,\ldots,k-i+1\}} A[j]$.

Invarijanta spoljašnje petlje (1) je :

niz $B[1..i]$ je rastući i $\min_{j\in\{1,2,\ldots,k-i\}} A[j] > \max_{j\in\{1,2,\ldots,i\}} B[j]$ i skup $\bigcup_{j=1}^{i}\{B[j]\} \cup \bigcup_{j=1}^{k-i}\{A[j]\}$ je konstantan i jednak skupu elemenata početnog niza A).

Pre ulaska u petlju (1) niz B je prazan, pa je i rastući. Niz B je prazan, pa možemo smatrati da je minimalni element niza A veći od svakog, pa i od maksimalnog elementa niza B. Važi i $\bigcup_{j=1}^{i}\{B[j]\} \cup \bigcup_{j=1}^{k-i}\{A[j]\} = \bigcup_{j=1}^{0}\{B[j]\} \cup \bigcup_{j=1}^{k-0}\{A[j]\} = \bigcup_{j=1}^{k}\{A[j]\}$, pa je ova unija upravo skup elemenata polaznog niza A. Dakle, tvrdjenje invarijante važi pre ulaska u petlju.

Pretpostavimo da tvrdjenje invarijante važi pre nekog izvršenja bloka naredbi u okviru petlje (za vrednost i) i dokažimo da važi i nakon tog izvršenja (za $i' = i + 1$). Pretpostavimo, dakle, da važi: niz $B[1..i]$ je rastući i $\min_{j\in\{1,2,\ldots,k-i\}} A[j] > \max_{j\in\{1,2,\ldots,i\}} B[j]$ i skup $\bigcup_{j=1}^{i}\{B[j]\} \cup \bigcup_{j=1}^{k-i}\{A[j]\}$ je je jednak skupu elemenata polaznog niza A. Na osnovu svojstava petlje (2), važi $A[m'] = \min_{j\in\{1,2,\ldots,k-i'+1\}} A[j]$ (gde je m' nova vrednost promenljive `m` i $i' = i + 1$ nova vrednost promenljive `i`). Dakle, važi $A[m'] = \min_{j\in\{1,2,\ldots,k-i\}} A[j]$, pa je na osnovu induktivne pretpostavke $A[m'] = \min_{j\in\{1,2,\ldots,k-i\}} A[j] > \max_{j\in\{1,2,\ldots,i\}} B[j]$, odakle sledi da je $B[i + 1] = B[i'] = A[m'] > \max_{j\in\{1,2,\ldots,i\}} B[j]$, tj. niz $B[1..i']$ je rastući. Nakon prethodne iteracije važilo je $\min_{j\in\{1,2,\ldots,k-i\}} A[j] > \max_{j\in\{1,2,\ldots,i\}} B[j]$, pa nakon komande `B[i]:=A[m];` element $B[i'] = B[i + 1]$ dobija vrednost A[m'] koja je bila minimum niza $A[1..k-i'+1]$ i koja postaje maksimum niza $B[1..i']$. Nakon komande `A[m]:=A[k-i+1];` iz niza $A[1..k - i' + 1]$ je izbačen minimalni element $A[m']$ (što nećemo posebno dokazivati), pa je minimum novodobijenog niza $A[1..k - i']$ veći (strogo veći, jer su svi elementi zadatog niza različiti) od minimuma niza $A[1..k - i]$ u prethodnoj iteraciji. Dakle, važi $\min_{j\in\{1,2,\ldots,k-i'\}} A[j] > \max_{j\in\{1,2,\ldots,i'\}} B[j]$.

Pored toga, nakon komande `B[i]:=A[m];` važi i $\bigcup_{j=1}^{i'}\{B[j]\} = \bigcup_{j=1}^{i}\{B[j]\} \cup \{B[i']\} = \bigcup_{j=1}^{i}\{B[j]\} \cup \{A[m']\}$ i nakon petlje `for l:=m to k-i do A[l]:=A[l+1];` iz niza $A[1..k-i'+1]$ je izbačen element $A[m']$, pa je skup $\bigcup_{j=1}^{k-i'}\{A[j]\}$ jednak razlici skupa $\bigcup_{j=1}^{k-i}\{A[j]\}$ nakon prethodne iteracije i skupa $\{A[m']\}$. Na osnovu induktivne pretpostavke, nakon prethodne iteracije važilo je da je skup $\bigcup_{j=1}^{i}\{B[j]\} \cup \bigcup_{j=1}^{k-i}\{A[j]\}$ jednak polaznom skupu elemenata niza A, pa (kako su svi elementi polaznog niza različiti) važi i $(\bigcup_{j=1}^{i}\{B[j]\} \cup \{A[m']\}) \cup (\bigcup_{j=1}^{k-i}\{A[j]\} \setminus \{A[m']\}) = (\bigcup_{j=1}^{i}\{B[j]\} \cup \{B[i']\}) \cup (\bigcup_{j=1}^{k-i}\{A[j]\} \setminus \{A[m']\}) = \bigcup_{j=1}^{i'}\{B[j]\} \cup (\bigcup_{j=1}^{k-i}\{A[j]\} \setminus \{A[m']\})$, tj. taj skup je jednak skupu $\bigcup_{j=1}^{i'}\{B[j]\} \cup \bigcup_{j=1}^{k-i'}\{A[j]\}$ nakon tekuće iteracije, što dokazuje tvrdjenje invarijante.

U svakoj iteraciji vrednost promenljive `i` povećava se za 1, pa ona u konačnom broju koraka dostiže vrednost k i tada algoritam završava rad (dakle, algoritam terminira). Tada važi $i = k$, pa na osnovu svojstava invarijante važi[2]: niz $B[1..k]$ je rastući i skup $\bigcup_{j=1}^{k}\{B[j]\} \cup \bigcup_{j=1}^{0}\{A[j]\}$ je jednak skupu elemenata početnog niza A, tj. niz $B[1..k]$ je rastući i skup $\bigcup_{j=1}^{k}\{B[j]\}$ je jednak skupu elemenata početnog niza A, što je i trebalo dokazati.