

Konstrukcija i analiza algoritama 2 (ispit, smer R, smer I)

1. Dat je niz brojeva -4 -1 -3 -4 -3 0 -1. Prikazati po koracima sortiranje ovog niza koristeći algoritam COUNTING SORT.

Resenje: Ako dodamo +5 svakom članu niza, onda novi niz A postaje 1 4 2 1 2 5 4

Primenimo algoritam:

COUNTING-SORT(A, B, k)

for $i \leftarrow 0$ to k do $C[i] \leftarrow 0$

for $j \leftarrow 1$ to $length[A]$ do $C[A[j]] \leftarrow C[A[j]] + 1$

for $i \leftarrow 1$ to k do $C[i] \leftarrow C[i] + C[i - 1]$

for $j \leftarrow length[A]$ downto 1 do

$B[C[A[j]]] \leftarrow A[j]$

$C[A[j]] \leftarrow C[A[j]] - 1$

Posle 2. petlje nakon učitavanja, evo kako izgleda niz A i niz broja pojava C:

A	1	4	2	1	2	5	4	C	2	2	0	2	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---

Posle 3. petlje nakon učitavanja, evo kako izgleda niz C(c[i] je broj elemenata manjih ili jednakih od i.):

C	2	4	4	6	7
---	---	---	---	---	---

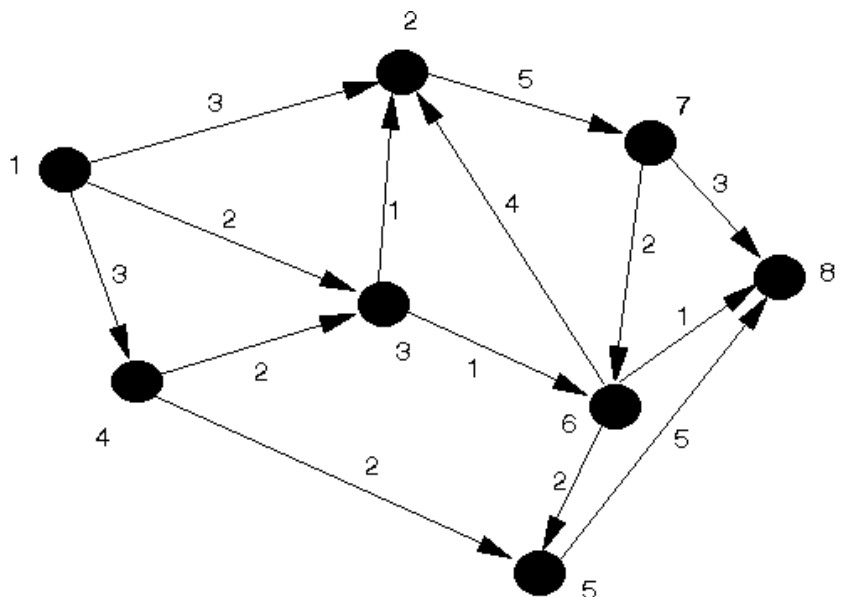
Posle 4. petlje nakon učitavanja, evo kako izgledaju nizovi B, C:

U ovoj petlji se formira niz b, tako što se, najpre, na c[i]-to mesto u niz b postavi taj broj a[i], pa pošto je ono sada popunjeno, c[i] se smanjuje za jedan. Postupak se ponavlja dok se ne formira ceo niz b.

B								C	2	4	4	6	7
B						4		C	2	3	4	5	7
B						4	5	C	2	3	4	5	6
B				2		4	5	C	2	3	4	5	6
B		1		2		4	5	C	1	3	4	5	6
B		1	2	2		4	5	C	1	2	4	5	6
B		1	2	2	4	4	5	C	1	2	4	4	6
B	1	1	2	2	4	4	5	C	0	2	4	4	6
B-5	-4	-4	-3	-3	-1	-1	0	C	0	2	4	4	6

2. Dat je usmereni graf $G = (V, E)$ sa dva istaknuta čvora 1 i 8, tako da je svakoj grani grafa dodeljen kapacitet.

Odrediti makimalni protok od čvora 1 do čvora 8 kroz datu mrežu i obrazložiti odgovor.



Resenje:

Ulazni graf mozemo predstaviti matricom

From \ To	Node1	Node2	Node3	Node4	Node5	Node6	Node7	Node8
Node1		3	2	3				
Node2							5	
Node3		1				1		
Node4			2		2			
Node5								5
Node6		4			2			1
Node7						2		3
Node8								

Izlazni rezultat se moze predstaviti matricom

	From	To	Net Flow		From	To	Net Flow
1	Node1	Node2	3	7	Node4	Node5	2
2	Node1	Node3	2	8	Node5	Node8	3
3	Node1	Node4	2	9	Node6	Node5	1
4	Node2	Node7	4	10	Node6	Node8	1
5	Node3	Node2	1	11	Node7	Node6	1
6	Node3	Node6	1	12	Node7	Node8	3
Total	Net Flow	From	Node1	To	Node8	=	7

Vidimo da maksimalni protok koji se moze poslati od cora 1 do cvora 8 ima vrednost $2+3+1+1+1+3=7$

3. Dat je skup tačaka $S = \{(6,3), (4,2), (4,1), (5,4), (7,7), (7,4), (2,5), (8,2)\}$. Korišćenjem Grejemovog algoritma nađite konveksni omotač za skup S.

REŠENJE: su 4 tacke skupa S

$(2,5), (4,1), (8,2), (7,7)$

4. Konstruisati CREW algoritam za sortiranje niza različitih brojeva x_1, x_2, \dots, x_n . Vremenska složenost treba da bude $O(\log n)$ na paralelnom CREW računaru sa zajedničkom memorijom i dovoljnim brojem procesora.

REŠENJE:

Pošto je na raspolaganju dovoljan broj procesora, a dozvoljeno je istovremeno čitanje sa iste lokacije, može se najpre formirati matrica A sa elementima

$A[i][j]=1$, ako $x_i < x_j$

$A[i][j]=0$, ako $x_i > x_j$

za šta je potrebno $O(1)$ paralelnih koraka.

Rang i_r elementa x_r (pozicija x_r u sortiranom redosledu) jednak je broju elemenata x_j manjih od x_r uvećanom za jedan, odnosno $s_r + 1$, gde je s_r zbir elemenata r -te vrste matrice A.

Pomoću n procesora zbrovi vrsta u A mogu se paralelno izracunati za $O(\log n)$ koraka (posebnim turnirom za svaku vrstu).

Zatim se u jednom paralelnom koraku x_r kopira na poziciju i_r izlaznog vektora, $1 \leq r \leq n$.

5. Za čvor binarnog stabla kaže se da je S-AVL čvor ako nije list i ako razlika visina njegovog levog i desnog podstabla je jednaka 0. Konstruisati algoritam čija vremenska složenost je linearna po broju čvorova binarnog stabla i koji u stablu označava sve S-AVL čvorove čiji niti jedan potomak nije S-AVL čvor.

REŠENJE:

Algoritam SAVL (stablo x)

ulaz: stablo x

izlaz: dubina stabla kao oznacen broj

/*

IDEJA : ako algoritam vrati dubinu <0, to znači da se u stablu kao potomak pojavljuje SAVL cvor
ako algoritam vrati dubinu 0, to znači da stablo je prazno
algoritam vrati dubinu >0, to znači da SAVL čvor jos nije nađen

NAPOMENA:

1. Ako su dubine levog i desnog podstabla jednake i ako ni u levom ni u desnom podstablu se ne nalazi SAVL čvor (jer je vratena dubina >0), tada se taj čvor markira, jer je SAVL čvor (npr. sa x->oznaka=1;). U tom slucaju vraća se negativna dubina stabla.

2. Ako nisu dubine levog i desnog podstabla jednake ili ako se u levom ili u desnom podstablu nalazi neki SAVL čvor (jer je vratena dubina <0), tada se vraća dubina drveta sa predznakom, zavisno od toga da li je nađen SAVL čvor ili ne. */

{

if (x==NULL) return 0;

else

{

levo=SAVL(x->levo);

desno=SAVL(x->desno);

if (levo >0 && desno >0)

if (levo==desno) /*situacija 1*/

{ x->oznaka=1; return -(levo+1);}

else return max(levo, desno) +1;

else return -(1+ max (abs(levo), max(abs(desno)));

}

Složenost algoritma:

Svako čvor se jednom obrađuje u const vremenu, te važi:

za $n=1$, $T(n) = c1$ (const za obradu stabla sa jednim čvorom)

za $n>1$, $T(n) = n * c1 * c2$ ($c2$ je cena vremena koje obuhvata rekurzivni poziv SAVL algoritma i povratka) =>
 $T(n) = O(n)$ q.e.d.