

ANALIZA I DIZAJN ALGORITAMA II

zadaci sa vežbi

Vesna Pavlović

28. decembar 2010.

10 Paralelni algoritmi

1. Neka je $*$ proizvoljna asocijativna binarna operacija (tj. važi: $x * (y * z) = x * (y * z)$) za proizvoljne x, y, z) i neka je dat niz brojeva x_1, x_2, \dots, x_n .

Izračunati proizvode $x_1 * x_2 * \dots * x_k$ za $k = 1, 2, \dots, n$.

Ovaj problem nazivamo *paralelni problem prefiksa*.

Algoritam Paralelni_prefiks_1(x, n)

Ulaz x (niz od n elemenata)

Izlaz x (niz čiji i -ti element sadrži i -ti prefiks)

begin

 PP_1(1, n);

end

Procedure PP_1($Levi, Desni$)

begin

 if $Desni - Levi = 1$ then

$x[Desni] := x[Levi] * x[Desni]$;

 else

$Srednji := (Levi + Desni - 1)/2$;

 do in parallel

 PP_1($Levi, Srednji$);

 PP_1($Srednji + 1, Desni$);

 for $i := Srednji + 1$ to $Desni$ do in parallel

$x[i] := x[Srednji] * x[i]$;

end

Algoritam Paralelni_prefiks_2(x, n)

Ulaz x (niz od n elemenata)

Izlaz x (niz čiji i -ti element sadrži i -ti prefiks)

begin

 PP_2(1);

end

Procedure PP_2($Korak$)

begin

 if $Korak = n/2$ then

$x[n] := x[n/2] * x[n]$;

 else

 for $i := 1$ to $n/(2 * Korak)$ do in parallel

$x[2 * i * Korak] := x[(2 * i - 1) * Korak] * x[2 * i * Korak]$;

 PP_2($2 * Korak$);

 for $i := 1$ to $n/(2 * Korak) - 1$ do in parallel

$x[(2 * i + 1) * Korak] := x[2 * i * Korak] * x[(2 * i + 1) * Korak]$;

end

2. Konstruisati algoritam za računar sa zajedničkom memorijom i n procesora, model EREW, koji na lokaciju $A[i]$ vektora A smešta vrednost $A[0] + i$, $1 \leq i < n$.
3. Konstruisati EREW algoritam za računar sa zajedničkom memorijom i n procesora koji na lokaciju $A[i]$, $i = 1, \dots, n$ vektora A smešta vrednost $i!$. Složenost predloženog algoritma treba da bude $O(\log n)$, a efikasnost $O(1)$.
4. Paralelizovati Hornerovu šemu za izračunavanje vrednosti polinoma stepena n po modelu EREW, tako da složenost, odnosno efikasnost algoritma, budu redom $O(\log n)$, odnosno $O(1)$.
5. Zadati je vektor čiji su elementi slogovi. Svaki slog sadrži neke podatke i Bulovu promenljivu *Oznaka*. Cilj je prepakovati na početak vektora sve slogove kojima polje *Oznaka* ima vrednost 1. Redosled premeštanih slogova treba da ostane nepromenjen.

Konstruisati EREW algoritam složenosti $T(n, n) = O(\log n)$ za rešavanje ovog problema.

6. *Kronekerov proizvod* vektora $A[0], A[1], \dots, A[n-1]$ i $B[0], B[1], \dots, B[m-1]$ se definiše kao vektor $C[0], C[1], \dots, C[nm-1]$ sa elementima:

$$C[km+r] = A[k]B[r], \quad 0 \leq k \leq n-1, \quad 0 \leq r \leq m-1.$$

Drugim rečima C sadrži $n \times m$ matricu čiji je (i, j) -ti element $A[i]B[j]$.

Konstruisati paralelni EREW algoritam za izračunavanje C pomoću p procesora za vreme $O(mn \log p/p)$. Na raspolaganju je memorijski prostor veličine $O(mn)$, i može se pretpostaviti da su m, n i p stepeni dvojke.

7. Data je povezana lista od n elemenata koji su smešteni u niz A dužine n (redosled elemenata liste nezavisan je od redosleda u nizu). Rang elementa u povezanoj listi definiše se kao rastojanje elementa od kraja liste (prvi element ima rang n , drugi $n-1, \dots$).

Izračunati rangove svih elemenata liste.

Algoritam Rangovi(N)

Ulaz N (niz od n elemenata)

Izlaz R (rangovi svih elemenata u nizu)

```

begin
   $D := 1$ ;
  do in parallel
     $R[i] := 0$ ;
    if ( $N[i] = NIL$ ) then  $R[i] := 1$ ;
    while ( $R[i] = 0$ ) do
      if ( $R[N[i]] \neq 0$ ) then
         $R[i] := D + R[N[i]]$ ;
      else
         $N[i] := N[N[i]]$ ;
         $D := D * 2$ ;
    end
  end

```

8. Dokazati da je za izvršavanje algoritma *Rangovi* dovoljan model EREW.
9. Data je povezana lista čiji su elementi (proizvoljnim redosledom) smešteni u vektor. Konstruisati efikasan paralalni algoritam za formiranje povezane liste od istih elemenata u obrnutom redosledu (ne premeštajući nijedan element). Može se pretpostaviti da je na raspolaganju dovoljan memorijski prostor za dopunske pokazivače.

10. Data je povezana lista čiji su elementi proizvoljnim redosledom smešteni u vektor. Neka je $F[i]$ Bulova promenljiva pridružena i -tom elementu liste. Konstruisati EREW algoritam složenosti $T(n, n) = O(\log n)$ za formiranje druge povezane liste, koja se sastoji samo od onih elemenata za koje je $F[i] = 1$, i to istim onim redosledom kojim se elementi pojavljuju u povezanoj listi.
11. Numerisati čvorove datog stabla EREW algoritmom prema dolaznoj numeraciji čvorova.
12. Konstruisati EREW algoritam koji izračunava broj potomaka za svaki čvor u stablu.