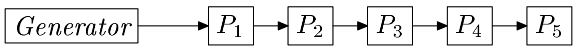
Ćwiczenia 6

Sortowanie n liczb na linii procesorów (z generatorem)

Dane: Dana linia n procesorów, procesor nr 1 otrzymuje liczby do posortowania (po jednej w każdym kroku).



Wynik: Linia procesorów powinna zawierać uporządkowane wartości (po jednej w każdym procesorze) — 1. procesor powinien zawierać najmniejszą wartość, n–ty procesor największą.

- każdy procesor działa wg. następującego schematu — jeśli otrzymuje liczbę z lewej strony, to:
  - jeśli nie ma zapisanej u siebie żadnej liczby, to ją zapamiętuje,
  - jeśli pamięta inną liczbę, to porównuje ją z otrzymaną, sobie zostawia mniejszą, a większą wysyła do prawego sąsiada.

Sortowanie n liczb na linii procesorów

Dane: Dana linia n procesorów, Każdy procesor ma przypisaną pewną wartość.
Wynik: Linia procesorów powinna zawierać uporządkowane wartości (po jednej w każdym procesorze) — 1. procesor powinien zawierać najmniejszą wartość, n–ty procesor największą.

Można rozwiązać ten problem symulując działanie sieci sortującej Odd–Even:

1. procesory o numerach $2i-1$ i $2i$ wymieniają się swoimi wartościami, procesor $2i-1$ zostawia sobie mniejszą z wartości (z zapisanej u siebie i otrzymanej), procesor $2i$ — większą.
2. procesory o numerach $2i$ i $2i+1$ wymieniają się swoimi wartościami, procesor $2i$ zostawia u siebie mniejszą z wartości, $2i+1$ większą.
3. kroki 1 i 2 należy wykonać $n/2$ razy

Generowanie liczb pierwszych na linii procesorów

Dane: Dana linia n procesorów.
Wynik: Procesory w linii mają zapisane kolejne liczby pierwsze.

---

1. 1. procesor generuje kolejne liczby naturalne zaczynając od 2, i przesyła je do swojego prawego sąsiada. Zatrzymuje się, gdy dostanie od prawego sąsiada specjalny sygnał.
2. pozostałe procesory, gdy dostaną liczbę od swojego lewego sąsiada, to:
   - jeśli jest to pierwsza liczba jaką dostały, to ją zapamiętują,

- wpp, jeśli otrzymana liczba nie dzieli się bez reszty przez pamiętaną, to jest wysyłana do prawego sąsiada.

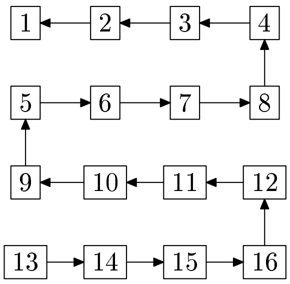
3. jeśli procesor (o numerze > 1) otrzyma sygnał od prawego sąsiada to przesyła go dalej do lewego sąsiada.

4. n. procesor, zaraz po tym jak otrzyma pierwszą liczbę od lewego sąsiada, wysyła mu specjalny sygnał stopu.

Shearsort

Dane: Dana siatka $n \times n$ procesorów, Każdy procesor ma przypisaną pewną wartość.

Wynik: Procesory zawierają posortowane wartości (rozważamy procesory w porządku "wężowym" — przedstawionym na rysunku)



Algorytm:

for $i = 1..2\log n + 1$ **do**
  if $odd(i)$ **then**
    sortuj wiersze w porządku wężowym
      (tzn. parzyste wiersze — malejąco, nieparzyste — rosnąco)
  else
    sortuj kolumny

Dowód poprawności:

Ponieważ algorytm *Shearshort* jest sztywny, stąd wystarczy pokazać, że poprawnie sortuje macierze zero–jedynkowe. Dowód opiera się na zliczaniu *brudnych* wierszy (tzn. takich które zawierają jednocześnie 0 i 1), podczas kolejnych kroków algorytmu. Na początku działania algorytmu takich wierszy może być n. Jedyną operacją zmieniającą liczbę brudnych wierszy jest sortowanie kolumn.

Gdy zgrupuje się brudne wiersze tablicy w pary kolejnych elementów, to okazuje się, że po ich posortowaniu, z każdej pary jeden z wierszy zmieni status na czysty. Tak więc z każdym krokiem liczba brudnych wierszy maleje (co najmniej) 2 razy. Zatem po $\log n$ krokach zostanie dokładnie jeden brudny wiersz.