

Algorytmy Równoległe i Rozproszone

Część II - Sieci porównujące

Łukasz Kuszner
pokój 209, WETI
<http://www.sphere.pl/~kuszner/>
kuszner@sphere.pl

Oficjalna strona wykładu
<http://www.sphere.pl/~kuszner/ARiR/>

Wykład 30 godzin, Ćwiczenia 15 godzin

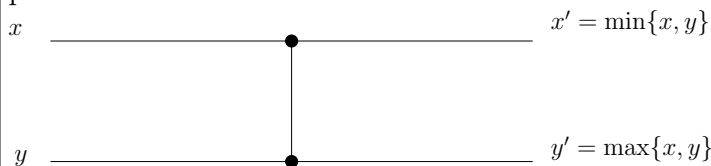
2005/06

Spis treści

1 Sieci porównujące	2
1.1 Głębokość sieci	2
1.2 Sieci sortujące	3
2 Zasada zero-jedynkowa	4
3 Sieci bitoniczne	5
4 Sieć sortująca	7

1 Sieci porównujące

Sieć porównująca składa się tylko z przewodów i komparatorów.



Zakładamy, że komparator działa w czasie $O(1)$, to znaczy czas pomiędzy pojawieniem się danych na wejściu (na rysunku po lewej stronie), a pojawieniem się wyników na wyjściu (na rysunku po prawej stronie) jest stały. Przyjmujemy, że sieć ma n wejść a_1, a_2, \dots, a_n i n wyjść b_1, b_2, \dots, b_n , a graf sieci jest acykliczny.

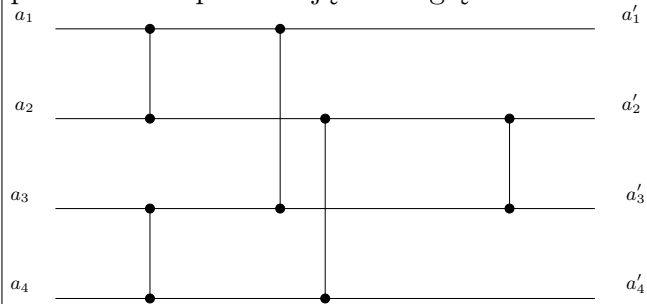
Notatki

1.1 Głębokość sieci

Wejście sieci ma głębokość 0. Jeśli wejścia komparatora mają głębokość d_x i d_y , to wyjście ma głębokość $\max(d_x, d_y) + 1$. *Głębokość sieci* definiujemy jako największą głębokość wyjścia. Głębokość identyfikujemy z czasem potrzebnym na posortowanie wszystkich elementów.

Notatki

poniższa sieć porównująca ma głębokość 3.



1.2 Sieci sortujące

Sieć sortująca, to sieć porównująca, której ciąg wyjściowy jest niemalejący dla każdego ciągu wejściowego.

Ćwiczenie 1

Wykaż, że sieć sortująca o n wejściach ma głębokość co najmniej $\lg n$.

Notatki

2 Zasada zero-jedynkowa

Lemat 1 *Jeśli sieć porównująca dla ciągu wejściowego $a = (a_1, a_2, \dots, a_n)$ wyznacza ciąg wyjściowy $b = (b_1, b_2, \dots, b_n)$, to dla dowolnej funkcji niemalejącej ta sama sieć z ciągiem wejściowym $f(a) = (f(a_1), f(a_2), \dots, f(a_n))$ wyznacza ciąg wyjściowy $f(b) = (f(b_1), f(b_2), \dots, f(b_n))$.*

Wskazówka 1: Rozważ pojedynczy komparator. Z monotoniczności f wynika, że $\max\{f(x), f(y)\} = f(\max\{x, y\})$ i $\min\{f(x), f(y)\} = f(\min\{x, y\})$

Wskazówka 2: Zastosuj indukcję na głębokość sieci.

Wskazówka 3: zob. Cormen str. 718

Ćwiczenie 2

Przeprowadź pełny dowód lematu 1.

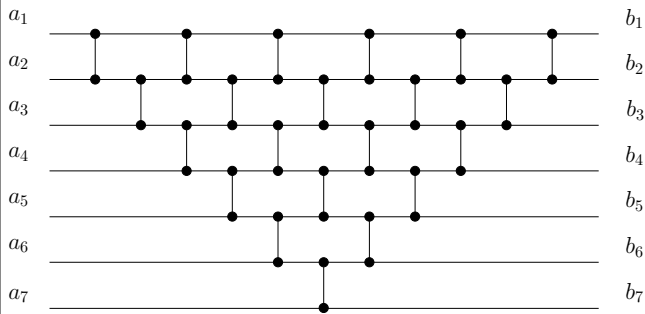
Notatki

Twierdzenie 2 *Jeśli sieć porównująca o n wejściach sortuje poprawnie wszystkie 2^n ciągi zer i jedynek, to sortuje poprawnie dowolne ciągi liczb.*

Dowód: Dowód nie wprost. Przypuśćmy, że sieć sortuje poprawnie wszystkie ciągi zer i jedynek, ale istnieje ciąg liczb $a = (a_1, a_2, \dots, a_n)$, dla którego wynik nie jest posortowany poprawnie. Istnieją elementy a_i, a_j takie, że $a_i < a_j$ oraz w ciągu wyjściowym a_j występuje przed a_i . Niech $f(x) = 0$, jeśli $x \leq a_i$ i $f(x) = 1$, jeśli $x > a_i$. Funkcja f jest niemalejąca. Z lematu 1 wnioskujemy, że jeśli na wejściu znajduje się ciąg $f(a) = (f(a_1), f(a_2), \dots, f(a_n))$, to na wyjściu wartość $f(a_j) = 1$ znajduje się przed $f(a_i) = 0$, sprzeczność z założeniem o poprawnym sortowaniu ciągów zero-jedynkowych. \square

Notatki

Rodzina sieci opartych na algorytmie sortowania przez wstawianie.



Ćwiczenie 3

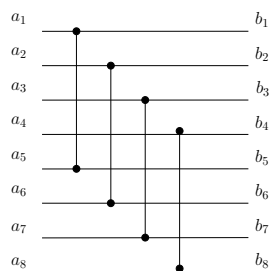
Jaka jest głębokość takiej sieci w zależności od liczby wejść?

Notatki

3 Bitoniczne sieci sortujące

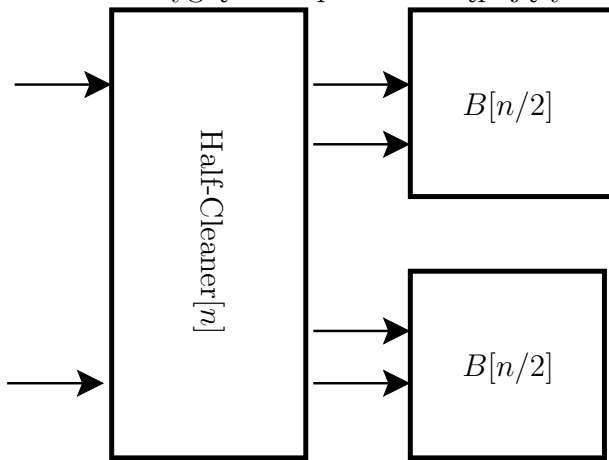
Ciągiem bitonicznym nazywamy ciąg który można przeciąć na dwie części: pierwszą niemalejącą i drugą nierosnącą lub pierwszą nierosnącą i drugą niemalejącą. Zero-jedynkowe ciągi bitoniczne mają postać $0^i 1^j 0^k$ albo $1^i 0^j 1^k$.

Elementem łączyszczą-cym (Half-Cleaner[n]), dla n parzystych nazywamy sieć porównującą o głębokości 1 złożoną z $n/2$ komparatorów, z których każdy łączy wejście o numerze i z wejściem $i + n/2$, dla $i = 1, 2, \dots, n/2$.



Notatki

Sieć sortującą ciągi bitoniczne $B[n]$ budujemy w następujący sposób. Dla $n = 1$ sieć jest pusta, dla $n = 2$ sieć składa się z jednego komparatora, natomiast dla $n > 2$ sieć wygląda w sposób następujący:



Notatki

Ćwiczenie 4

Narysuj sieć $B[8]$.

Ćwiczenie 5

Jak skonstruować sieć bitoniczną, gdy n nie jest potęgą 2?

Ćwiczenie 6

Uzasadnij, że jeśli każdy element jednego z dwóch zero-jedynkowych ciągów bitonicznych jest nie większy niż każdy element drugiego ciągu, to jeden z ciągów składa się z samych 0, albo z samych 1.

Ćwiczenie 7

Przeczytaj ze zrozumieniem pełen dowód poprawności działania sieci $B[n]$, (Cormen str. 721) i odtwórz go nie zaglądając do książki.

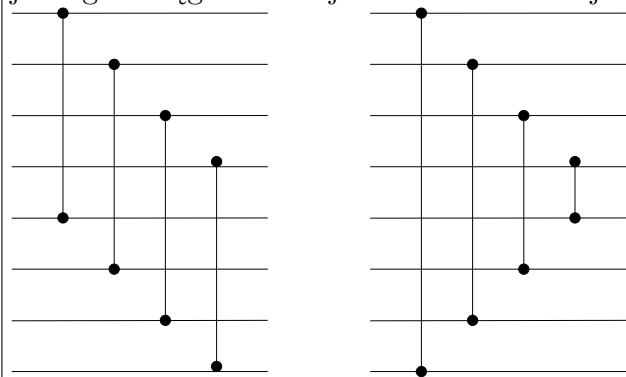
Notatki

4 Sieć sortująca o głębokości $\log^2 n$

Pokażemy teraz jak skonstruować sieć opartą na algorytmie `mergesort`. Do tego celu użyjemy pokazanych wcześniej sieci bitonicznych i sieci scalających. Konstrukcja sieci scalającej dwa posortowane ciągi w jeden opiera się na prostej obserwacji: jeśli w jednym z dwóch posortowanych ciągów odwrócimy porządek, to otrzymamy ciąg bitoniczny, a ciągi bitoniczne umiemy już sortować.

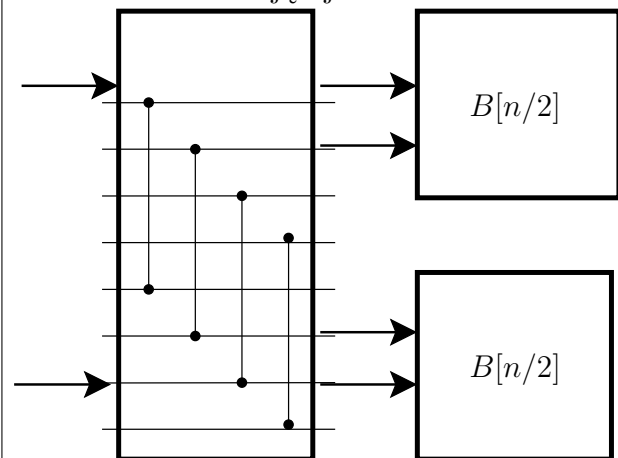
Notatki

Poniższy rysunek przedstawia jak odwrócić porządek jednego z ciągów na wejściu sieci bitoniczej.



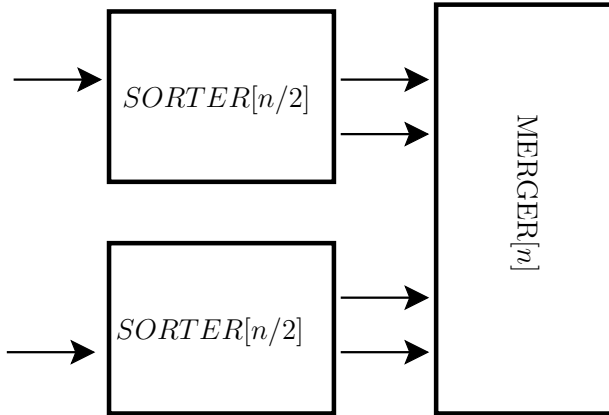
Notatki

Schemat sieci scalającej MERGER[n]



Notatki

Mając już wszystkie potrzebne elementy możemy zbudować sieć sortującą $SORTER[n]$



Notatki

Ćwiczenie 8

Narysuj sieć $SORTER[8]$.

Ćwiczenie 9

Wykaż, że głębokość sieci $SORTER[n]$ jest $\Theta(\log^2(n))$.

Ćwiczenie 10

Wykaż, że głębokość sieci $SORTER[n]$ jest równa $\lg n(\lg n + 1)/2$.

Notatki