Algorytmy Równoległe i Rozproszone
Część I - Wprowadzenie i 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 15 godzin, Projekt 15 godzin

2006/07

Zaliczenie
Plan wykładu
Literatura
Sieci porównujące
PRAM - przykład
Model rozproszony
Sieci porównujące
Zasada zero-jedynkowa
Sieci bitoniczne
Sieć sortująca

1. Zaliczenie

Na zaliczenie przedmiotu składają się następujące elementy:

- C - suma punktów uzyskanych na ćwiczeniach,

Zaliczenie
Plan wykładu
Literatura
Sieci porównujące
PRAM - przykład
Model rozproszony
Sieci porównujące
Zasada zero-jedynkowa
Sieci bitoniczne
Sieć sortująca

1. Zaliczenie

Na zaliczenie przedmiotu składają się następujące elementy:

- C - suma punktów uzyskanych na ćwiczeniach,
- K - maksymalna suma punktów uzyskana z kolokwiów, które odbędą się w trakcie trwania semestru.

Zaliczenie
Plan wykładu
Literatura
Sieci porównujące
PRAM - przykład
Model rozproszony
Sieci porównujące
Zasada zero-jedynkowa
Sieci bitoniczne
Sieć sortująca

1. Zaliczenie

Na zaliczenie przedmiotu składają się następujące elementy:

- C - suma punktów uzyskanych na ćwiczeniach,
- K - maksymalna suma punktów uzyskana z kolokwiów, które odbędą się w trakcie trwania semestru.
- D - suma punktów dodatkowych

Zaliczenie
Plan wykładu
Literatura
Sieci porównujące
PRAM - przykład
Model rozproszony
Sieci porównujące
Zasada zero-jedynkowa
Sieci bitoniczne
Sieć sortująca

Końcowy wynik S obliczamy według wzoru:

$$S = \left(\frac{C + D + K\}}{100} \right) \cdot 100\%$$

Ocenę z przedmiotu wyznaczamy w zależności od S w następujący sposób:

ocena	wynik
2	$\max\{K_1 + K_2, E\} < 100 \vee C < 100$
3	$S \geqslant 50\% \wedge S < 60\%$
3+	$S \geqslant 60\% \wedge S < 70\%$
4	$S \geqslant 70\% \wedge S < 80\%$
4+	$S \geqslant 80\% \wedge S < 90\%$
5	$S \geqslant 90\% \wedge S < 100\%$
5+	$S \geqslant 100\%$

2. Ogólny plan wykładu

- Wstęp

2. Ogólny plan wykładu

- Wstęp
- Sieci porównujące,

2. Ogólny plan wykładu

- Wstęp
- Sieci porównujące,
- Układy kombinacyjne,

Zaliczenie
Plan wykładu
Literatura
Sieci porównujące
PRAM - przykład
Model rozproszony
Sieci porównujące
Zasada zero-jedynkowa
Sieci bitoniczne
Sieć sortująca

2. Ogólny plan wykładu

- Wstęp
- Sieci porównujące,
- Układy kombinacyjne,
- Algorytmy w modelu PRAM,

2. Ogólny plan wykładu

- Wstęp
- Sieci porównujące,
- Układy kombinacyjne,
- Algorytmy w modelu PRAM,
- Algorytmy w modelu rozproszonym.

3. Literatura

- T. H. Cormen, C. E. Leiserson and R. L. Rivest, „Introduction to Algorithms", The MIT Press/McGraw-Hill Company, 1990 (wydanie polskie WNT).

Zaliczenie
Plan wykładu
Literatura
Sieci porównujące
PRAM - przykład
Model rozproszony
Sieci porównujące
Zasada zero-jedynkowa
Sieci bitoniczne
Sieć sortująca

3. Literatura

- T. H. Cormen, C. E. Leiserson and R. L. Rivest, „Introduction to Algorithms”, The MIT Press/McGraw-Hill Company, 1990 (wydanie polskie WNT).
- Raymond Greenlaw, H. James Hoover, Walter L. Ruzzo „Limits to Parallel Computation: P-Completeness Theory”, Oxford University Press, 1998.

Zaliczenie
Plan wykładu
Literatura
Sieci porównujące
PRAM - przykład
Model rozproszony
Sieci porównujące
Zasada zero-jedynkowa
Sieci bitoniczne
Sieć sortująca

3. Literatura

- T. H. Cormen, C. E. Leiserson and R. L. Rivest, „Introduction to Algorithms", The MIT Press/McGraw-Hill Company, 1990 (wydanie polskie WNT).
- Raymond Greenlaw, H. James Hoover, Walter L. Ruzzo „Limits to Parallel Computation: P-Completeness Theory", Oxford University Press, 1998.
- Hagit Attiya „Lecture Notes for Course Distributed Algorithms", 1994.

- C. Xavier, S. S. Iyengar, „Introduction to Parallel Algorithms”, Wiley-IEEE, 1998.

Zaliczenie
Plan wykładu
Literatura
Sieci porównujące
PRAM - przykład
Model rozproszony
Sieci porównujące
Zasada zero-jedynkowa
Sieci bitoniczne
Sieć sortująca

- C. Xavier, S. S. Iyengar, „Introduction to Parallel Algorithms”, Wiley-IEEE, 1998.
- Gerard Tel, „Introduction to Distributed Algorithms”, Cambridge University Press, 2nd edition, 2000.

Zaliczenie
Plan wykładu
Literatura
Sieci porównujące
PRAM - przykład
Model rozproszony
Sieci porównujące
Zasada zero-jedynkowa
Sieci bitoniczne
Sieć sortująca

- C. Xavier, S. S. Iyengar, „Introduction to Parallel Algorithms”, Wiley-IEEE, 1998.
- Gerard Tel, „Introduction to Distributed Algorithms”, Cambridge University Press, 2nd edition, 2000.
- Shlomi Dolev. Self-Stabilization. MIT Press, 2000.

- J. Jaja, „An Introduction to Parallel Algorithms”, Addison-Wesley, Reading, MA, 1992.

- J. Jaja, „An Introduction to Parallel Algorithms”, Addison-Wesley, Reading, MA, 1992.
- S.G. Akl, „The Design and Analysis of Parallel Algorithms”, Prentice-Hall, 1989.

Zaliczenie
Plan wykładu
Literatura
Sieci porównujące
PRAM - przykład
Model rozproszony
Sieci porównujące
Zasada zero-jedynkowa
Sieci bitoniczne
Sieć sortująca

- J. Jaja, „An Introduction to Parallel Algorithms”, Addison-Wesley, Reading, MA, 1992.
- S.G. Akl, „The Design and Analysis of Parallel Algorithms”, Prentice-Hall, 1989.
- Lecture Notes Repository: Uniwersytet w Padeborn.

- Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar „Introduction to Parallel Computing”, Addison Weslesy, 2003.

- Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar „Introduction to Parallel Computing”, Addison Weslesy, 2003.
- Hagit Attiya, Jennifer Welch „Distributed Computing: Fundamentals, Simulations, and Advanced Topics”, 2nd Edition.

- Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar „Introduction to Parallel Computing", Addison Weslesy, 2003.
- Hagit Attiya, Jennifer Welch „Distributed Computing: Fundamentals, Simulations, and Advanced Topics", 2nd Edition.
- Guy E. Blelloch, Bruce M. Maggs: Parallel Algorithms. The Computer Science and Engineering Handbook, 1997: 277-315.

Zaliczenie
Plan wykładu
Literatura
Sieci porównujące
PRAM - przykład
Model rozproszony
Sieci porównujące
Zasada zero-jedynkowa
Sieci bitoniczne
Sieć sortująca

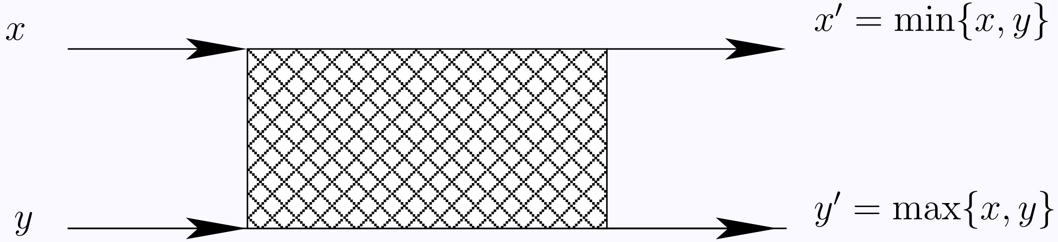
Zaliczenie
Plan wykładu
Literatura
Sieci porównujące
PRAM - przykład
Model rozproszony
Sieci porównujące
Zasada zero-jedynkowa
Sieci bitoniczne
Sieć sortująca

4. Sieci porównujące - przykład

Sieć porównująca składa się z połączonych ze sobą komparatorów.

4.1. Schematy komparatora

zwykły:



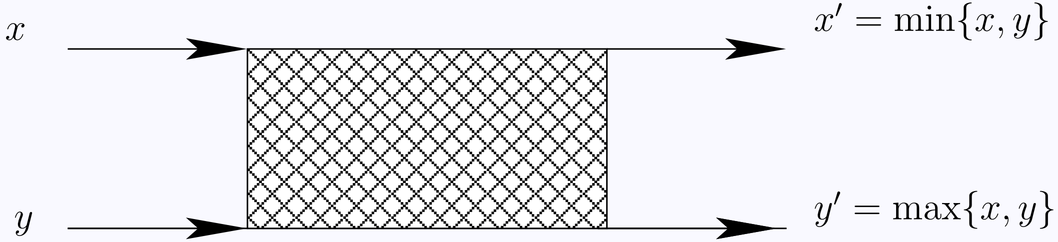
Zaliczenie
Plan wykładu
Literatura
Sieci porównujące
PRAM - przykład
Model rozproszony
Sieci porównujące
Zasada zero-jedynkowa
Sieci bitoniczne
Sieć sortująca

4. Sieci porównujące - przykład

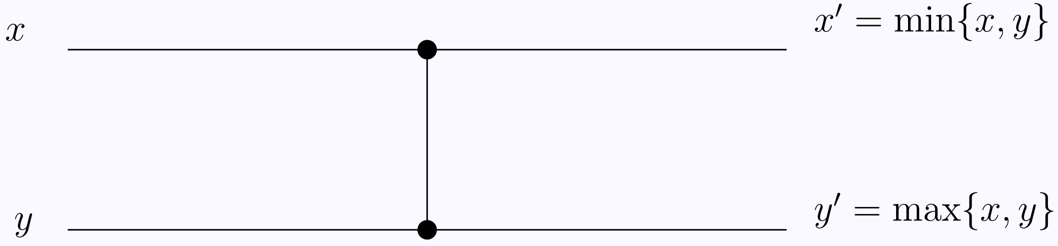
Sieć porównująca składa się z połączonych ze sobą komparatorów.

4.1. Schematy komparatora

zwykły:

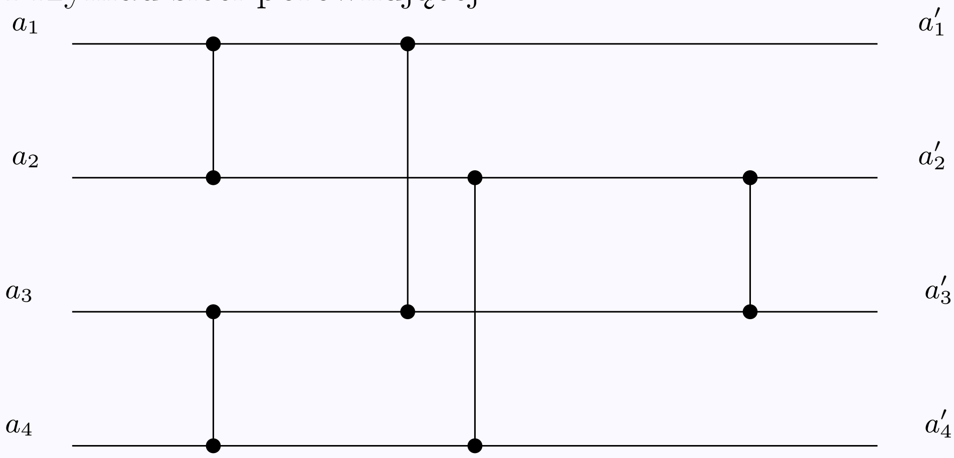


uproszczony:



Zaliczenie
Plan wykładu
Literatura
Sieci porównujące
PRAM - przykład
Model rozproszony
Sieci porównujące
Zasada zero-jedynkowa
Sieci bitoniczne
Sieć sortująca

Przykład sieci porównującej



Zaliczenie
Plan wykładu
Literatura
Sieci porównujące
PRAM - przykład
Model rozproszony
Sieci porównujące
Zasada zero-jedynkowa
Sieci bitoniczne
Sieć sortująca

Ćwiczenie 1

Sprawdź, że sieć porównująca z poprzedniego slajdu jest siecią sortującą.

5. PRAM - przykład

5.1. Iloczyn skalarny

We: Tablice współrzędnych $a[1:n]$ i $b[1:n]$
Wy: Liczba będąca iloczynem skalarnym wektorów a i b.

Algorytm 1: Iloczyn skalarny

```
for i = 1 to n in parallel do
    c_i = a_i * b_i
end for
p = n/2
while p > 0 do
    for i = 1 to p in parallel do
        c_i = c_i + c_{i+p}
    end for
    p = p/2
end while
```


Zaliczenie
Plan wykładu
Literatura
Sieci porównujące
PRAM - przykład
Model rozproszony
Sieci porównujące
Zasada zero-jedynkowa
Sieci bitoniczne
Sieć sortująca
Strona główna
Strona tytułowa
Strona 13 z 32
Powrót
Full Screen
Zamknij
Koniec

Przykład dla algorytmu, dla $n = 7$.

1	1	0	3	2	-1	1

$p = 4$ $+$

3	0	1	3	2	-1	1

$p = 2$

4	3	1	3	2	-1	1

$p = 1$

7	3	1	3	2	-1	1

Zaliczenie
Plan wykładu
Literatura
Sieci porównujące
PRAM - przykład
Model rozproszony
Sieci porównujące
Zasada zero-jedynkowa
Sieci bitoniczne
Sieć sortująca

6. Model rozproszony - przykład

6.1. Rozproszony algorytm dla problemu: „wybór lidera”

Przypuśćmy, że mamy rozproszony system asynchroniczny, w którym jednostki obliczeniowe tworzą pierścień.
Problem polega na skonstruowaniu algorytmu, który pozwoli wyróżnić jeden z procesorów. Oczekujemy, że po zakończeniu działania algorytmu dokładnie jeden procesor zasygnalizuje „jestem liderem”, a wszystkie pozostałe „nie jestem liderem”.

Zaliczenie
Plan wykładu
Literatura
Sieci porównujące
PRAM - przykład
Model rozproszony
Sieci porównujące
Zasada zero-jedynkowa
Sieci bitoniczne
Sieć sortująca

Przypuśćmy, że każdy z procesorów ma unikalny identyfikator. Procesory tworzą pierścień, możemy więc mówić o lewym i prawym sąsiedzie. Inicjalnie każdy z procesorów wysyła wiadomość do swojego lewego sąsiada, w której zapisuje swój identyfikator. Następnie każdy procesor odbierając wiadomość od prawego sąsiada przesyła ją dalej, jeśli zapisany na niej identyfikator jest większy od jego własnego identyfikatora i kasuje, jeśli jest mniejszy.
Procesor, który otrzymał z powrotem swój własny identyfikator ogłasza się liderem i wysyła wiadomość kończącą działanie algorytmu.

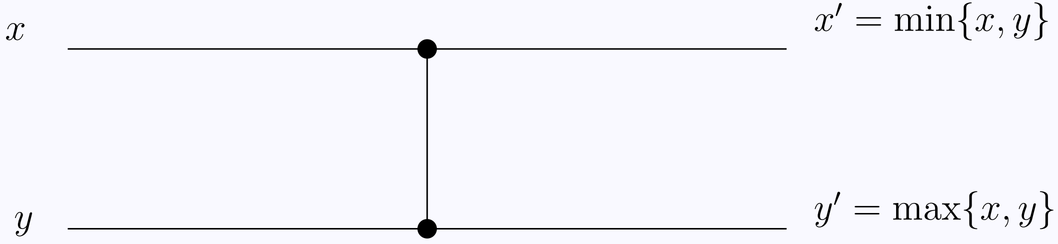
Zaliczenie
Plan wykładu
Literatura
Sieci porównujące
PRAM - przykład
Model rozproszony
Sieci porównujące
Zasada zero-jedynkowa
Sieci bitoniczne
Sieć sortująca

Ćwiczenie 2

Uzasadnij, że liczba wiadomości wysłanych w trakcie działania algorytmu jest $O(n^2)$.

7. Sieci porównujące

Sieć porównująca składa się tylko z przewodów i komparatorów.

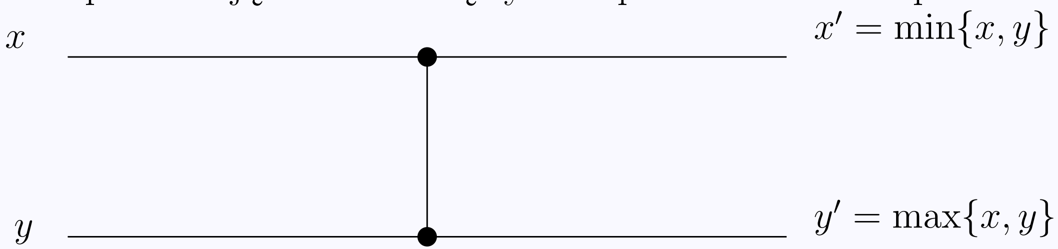


Zaliczenie
Plan wykładu
Literatura
Sieci porównujące
PRAM - przykład
Model rozproszony
Sieci porównujące
Zasada zero-jedynkowa
Sieci bitoniczne
Sieć sortująca

Strona główna
Strona tytułowa
Strona 18 z 32
Powrót
Full Screen
Zamknij
Koniec

7. 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 $a_1, a_2, \ldots, a_n$ i n wyjść $b_1, b_2, \ldots, b_n$, a graf sieci jest acykliczny.

Zaliczenie
Plan wykładu
Literatura
Sieci porównujące
PRAM - przykład
Model rozproszony
Sieci porównujące
Zasada zero-jedynkowa
Sieci bitoniczne
Sieć sortująca

Strona główna
Strona tytułowa
◀◀ ▶▶
◀ ▶
Strona 18 z 32
Powrót
Full Screen
Zamknij
Koniec

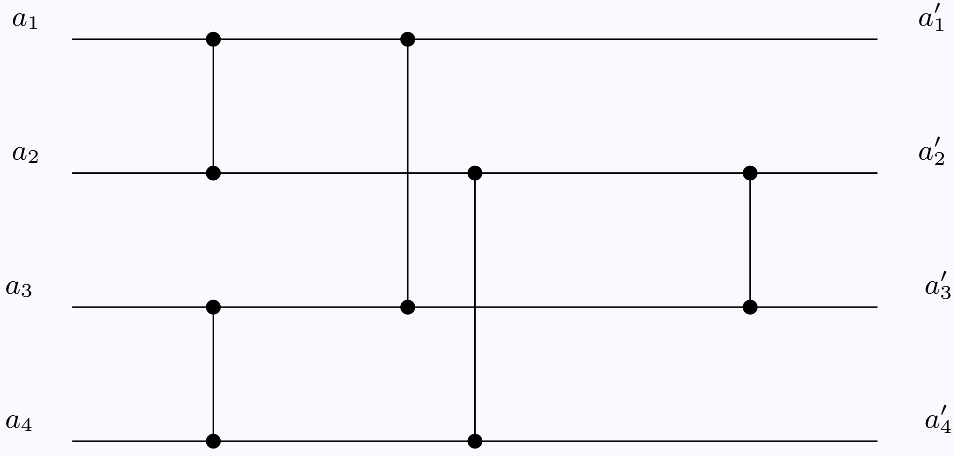
7.1. Głębokość sieci

Wejście sieci ma głębokość 0. Jeśli wejścia komparatora mają głębokość d_x i d_y, to wyjście ma głębokość $\max(d_x, d_y) + 1$.

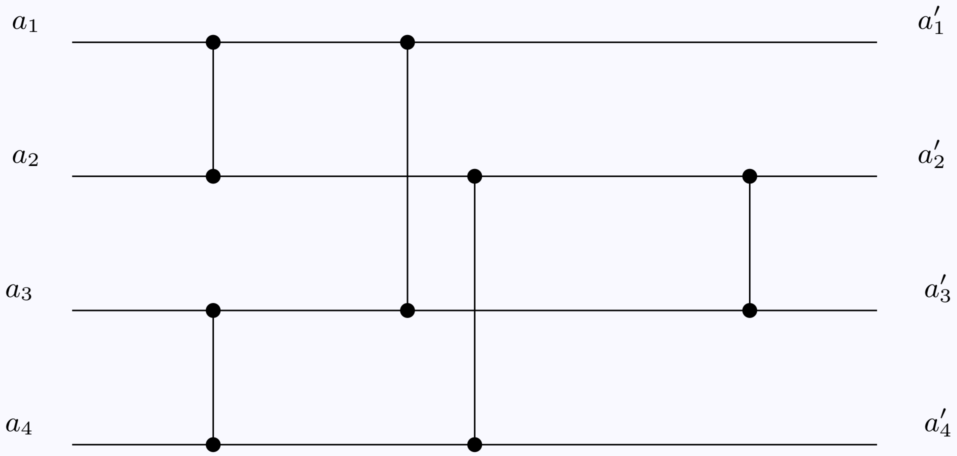
Zaliczenie
Plan wykładu
Literatura
Sieci porównujące
PRAM - przykład
Model rozproszony
Sieci porównujące
Zasada zero-jedynkowa
Sieci bitoniczne
Sieć sortująca

7.1. 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.



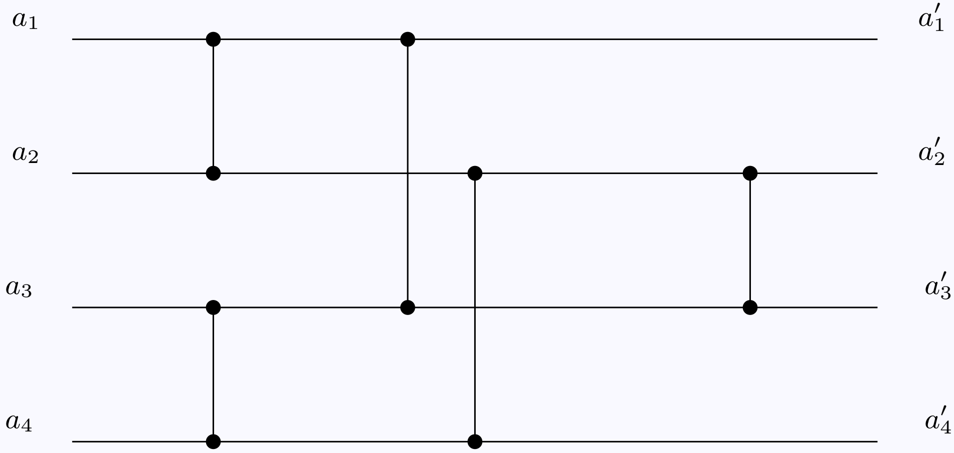
Zaliczenie
Plan wykładu
Literatura
Sieci porównujące
PRAM - przykład
Model rozproszony
Sieci porównujące
Zasada zero-jedynkowa
Sieci bitoniczne
Sieć sortująca



poniższa sieć porównująca ma głębokość 3.

7.2. Sieci sortujące

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



poniższa sieć porównująca ma głębokość 3.

7.2. Sieci sortujące

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

Ćwiczenie 3

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

Zaliczenie
Plan wykładu
Literatura
Sieci porównujące
PRAM - przykład
Model rozproszony
Sieci porównujące
Zasada zero-jedynkowa
Sieci bitoniczne
Sieć sortująca

8. Zasada zero-jedynkowa

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

Zaliczenie
Plan wykładu
Literatura
Sieci porównujące
PRAM - przykład
Model rozproszony
Sieci porównujące
Zasada zero-jedynkowa
Sieci bitoniczne
Sieć sortująca

8. Zasada zero-jedynkowa

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

Ćwiczenie 4

Przeprowadź pełny dowód lematu 1.

Zaliczenie
Plan wykładu
Literatura
Sieci porównujące
PRAM - przykład
Model rozproszony
Sieci porównujące
Zasada zero-jedynkowa
Sieci bitoniczne
Sieć sortująca

8. Zasada zero-jedynkowa

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

Ćwiczenie 4

Przeprowadź pełny dowód lematu 1.

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

Zaliczenie
Plan wykładu
Literatura
Sieci porównujące
PRAM - przykład
Model rozproszony
Sieci porównujące
Zasada zero-jedynkowa
Sieci bitoniczne
Sieć sortująca

8. Zasada zero-jedynkowa

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

Ćwiczenie 4

Przeprowadź pełny dowód lematu 1.

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.

Zaliczenie
Plan wykładu
Literatura
Sieci porównujące
PRAM - przykład
Model rozproszony
Sieci porównujące
Zasada zero-jedynkowa
Sieci bitoniczne
Sieć sortująca

8. Zasada zero-jedynkowa

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

Ćwiczenie 4

Przeprowadź pełny dowód lematu 1.

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

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

Zaliczenie
Plan wykładu
Literatura
Sieci porównujące
PRAM - przykład
Model rozproszony
Sieci porównujące
Zasada zero-jedynkowa
Sieci bitoniczne
Sieć sortująca

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 $a = (a_1, a_2, \ldots, a_n)$, 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.

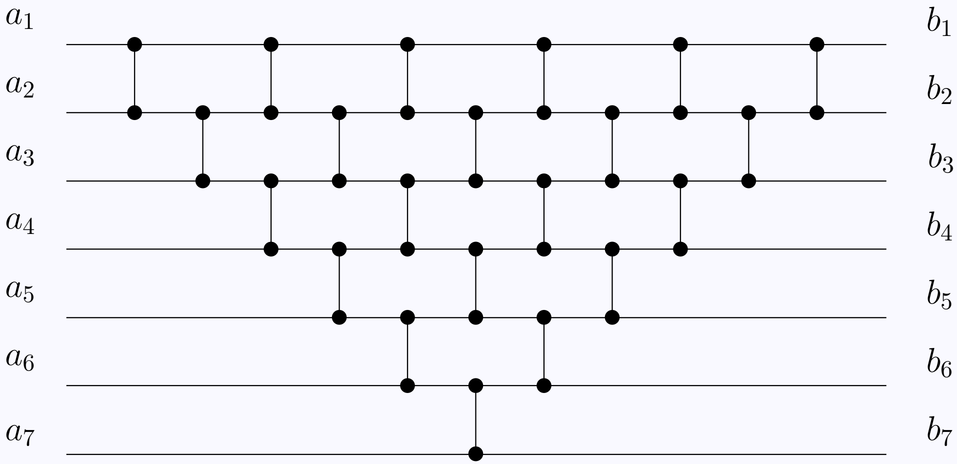
Zaliczenie
Plan wykładu
Literatura
Sieci porównujące
PRAM - przykład
Model rozproszony
Sieci porównujące
Zasada zero-jedynkowa
Sieci bitoniczne
Sieć sortująca

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

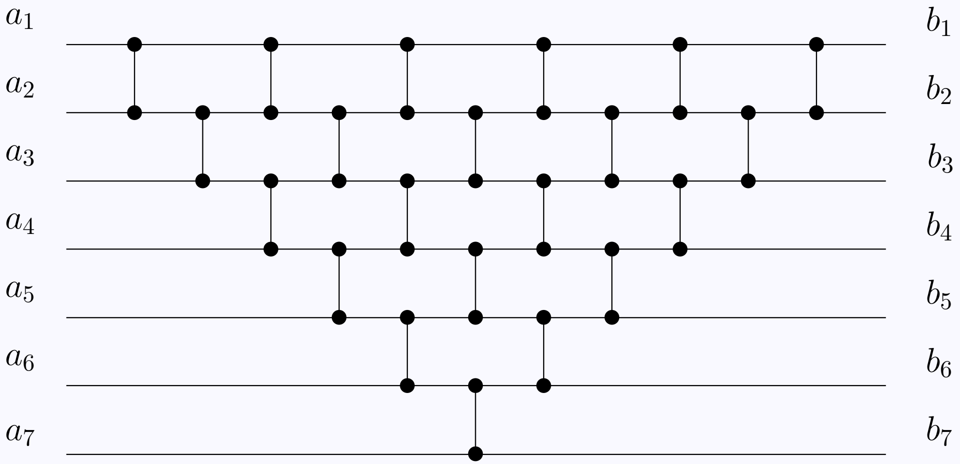
Dowód: Dowód nie wprost. Przypuśćmy, że sieć sortuje poprawnie wszystkie ciągi zer i jedynek, ale istnieje ciąg liczb $a = (a_1, a_2, \ldots, a_n)$, dla którego wynik nie jest posortowany poprawnie. Istnieją elementy a_i, a_j takie, że $a_i < a_j$ oraz w ciągu wyjściowym a_j występuje przed a_i. Niech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

Zaliczenie
Plan wykładu
Literatura
Sieci porównujące
PRAM - przykład
Model rozproszony
Sieci porównujące
Zasada zero-jedynkowa
Sieci bitoniczne
Sieć sortująca

Rodzina sieci opartych na algorytmie sortowania przez wstawianie.



Rodzina sieci opartych na algorytmie sortowania przez wstawianie.



Ćwiczenie 5

Jaka jest głębokość takiej sieci w zależności od liczby wejść?

9. Bitoniczne sieci sortujące

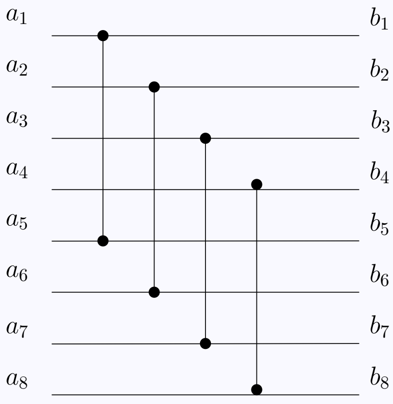
Ciągiem bitonicznym nazywamy ciąg, który można przeciąć na dwie części: pierwszą niemalejącą i drugą nierosnącą lub pierwszą nierosnącą i drugą niemalejącą. Zero-jedynkowe ciągi bitoniczne mają postać $0^i 1^j 0^k$ albo $1^i 0^j 1^k$.

Zaliczenie
Plan wykładu
Literatura
Sieci porównujące
PRAM - przykład
Model rozproszony
Sieci porównujące
Zasada zero-jedynkowa
Sieci bitoniczne
Sieć sortująca

9. Bitoniczne sieci sortujące

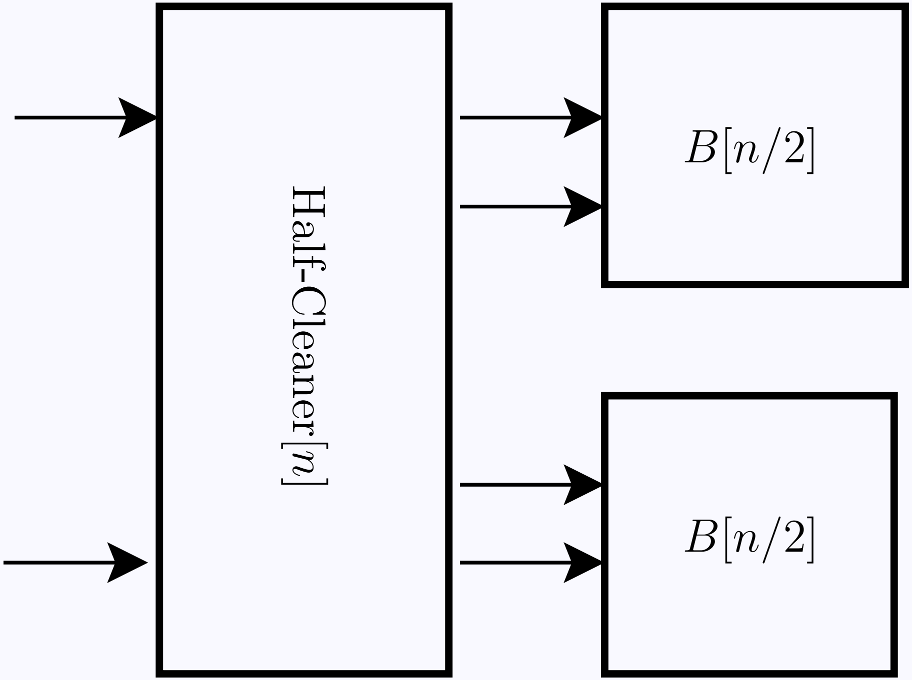
Ciągiem bitonicznym nazywamy ciąg, który można przeciąć na dwie części: pierwszą niemalejącą i drugą nierosnącą lub pierwszą nierosnącą i drugą niemalejącą. Zero-jedynkowe ciągi bitoniczne mają postać $0^i 1^j 0^k$ albo $1^i 0^j 1^k$.

Elementem półczyszczącym (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 $i = 1, 2, \ldots n/2$.



Zaliczenie
Plan wykładu
Literatura
Sieci porównujące
PRAM - przykład
Model rozproszony
Sieci porównujące
Zasada zero-jedynkowa
Sieci bitoniczne
Sieć sortująca

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:



Zaliczenie
Plan wykładu
Literatura
Sieci porównujące
PRAM - przykład
Model rozproszony
Sieci porównujące
Zasada zero-jedynkowa
Sieci bitoniczne
Sieć sortująca

Ćwiczenie 6

Narysuj sieć B[8].

Ćwiczenie 7

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

Ćwiczenie 8

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 9

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

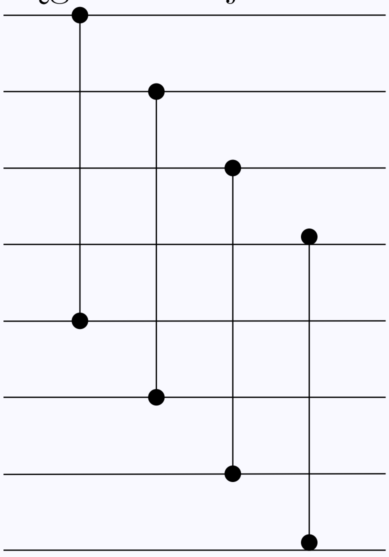
Zaliczenie
Plan wykładu
Literatura
Sieci porównujące
PRAM - przykład
Model rozproszony
Sieci porównujące
Zasada zero-jedynkowa
Sieci bitoniczne
Sieć sortująca

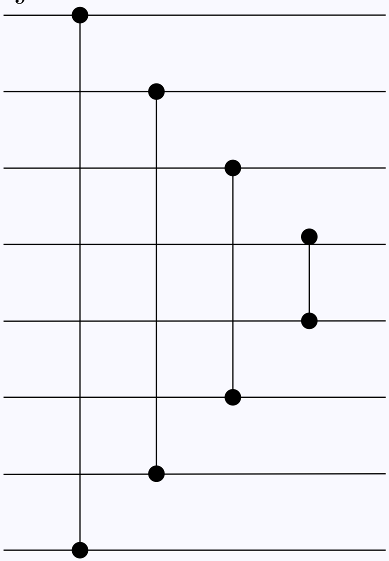
10. Sieć sortująca o głębokości $\log^2 n$

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.

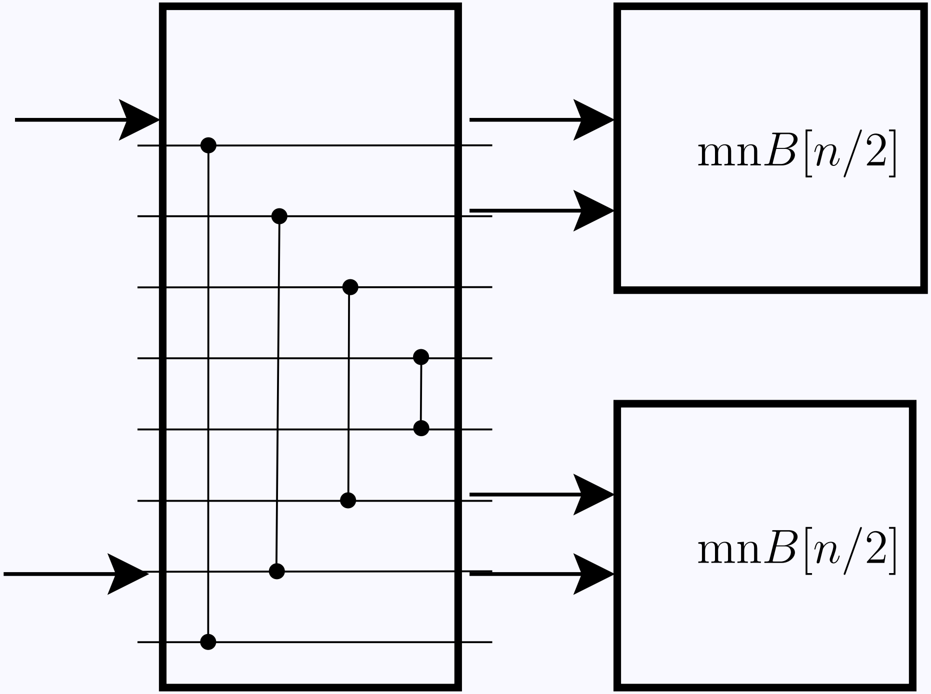
Zaliczenie
Plan wykładu
Literatura
Sieci porównujące
PRAM - przykład
Model rozproszony
Sieci porównujące
Zasada zero-jedynkowa
Sieci bitoniczne
Sieć sortująca

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



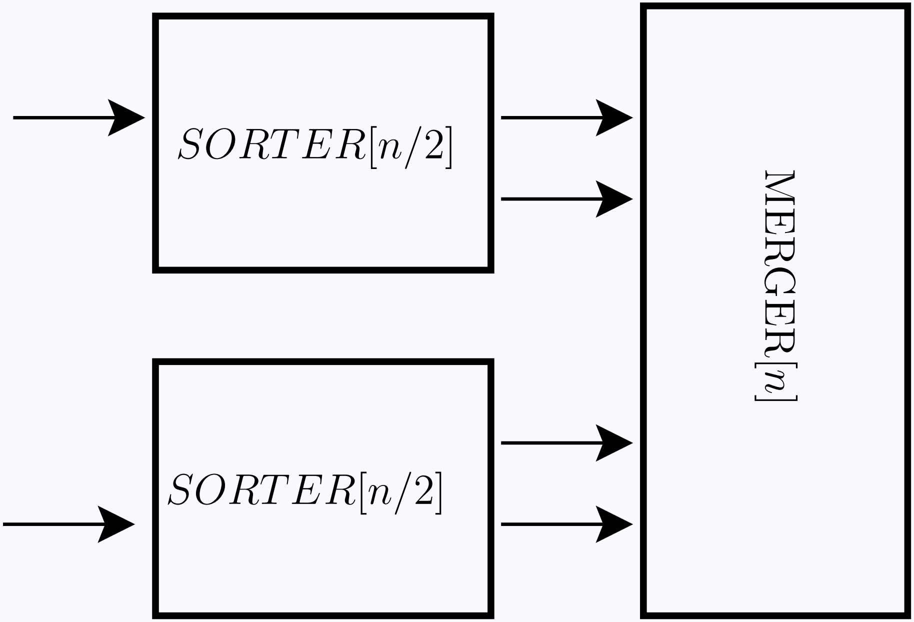


Schemat sieci scalającej `MERGER[`n`]`



Zaliczenie
Plan wykładu
Literatura
Sieci porównujące
PRAM - przykład
Model rozproszony
Sieci porównujące
Zasada zero-jedynkowa
Sieci bitoniczne
Sieć sortująca

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



Zaliczenie
Plan wykładu
Literatura
Sieci porównujące
PRAM - przykład
Model rozproszony
Sieci porównujące
Zasada zero-jedynkowa
Sieci bitoniczne
Sieć sortująca

Ćwiczenie 10

Narysuj sieć `SORTER[8]`.

Ćwiczenie 11

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

Ćwiczenie 12

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

Zaliczenie
Plan wykładu
Literatura
Sieci porównujące
PRAM - przykład
Model rozproszony
Sieci porównujące
Zasada zero-jedynkowa
Sieci bitoniczne
Sieć sortująca