



Parallele Sortierung

Björn Rathjen Patrick Winterstein
Freie Universität Berlin

Proseminar Algorithmen, SS14

Inhalt

Motivation

- Allgemein
- Bezug aufs Fach

Vorraussetzungen

- Komparator
- 0,1-Prinzip

Sortiernetzwerk

- Aufbau
- Sortieren im Sortiernetzwerk

Laufzeit

- Herleitung
- Vergleich mit Software sortieren

Fazit

- Geschwindigkeit vs Variabilität
- Hardwareaufwand vs Softwareaufwand

Zusammenfassung

Ausblick

- Hypercube
- Anhang

Motivation

Allgemein

Bezug aufs Fach

Vorraussetzungen

Sortiernetzwerk

Laufzeit

Fazit

Zusammenfassung

Ausblick

ist Basis für :

- ▶ Sortierung
- ▶ Suche

- ▶ Listen , Wörterbücher
- ▶ Datenbanken
- ▶ ...

Motivation

Vorraussetzungen

Komparator

0,1-Prinzip

Sortiernetzwerk

Laufzeit

Fazit

Zusammenfassung

Ausblick

abstrakter Aufbau

- ▶ 2 Eingänge
- ▶ vergleichender Baustein
- ▶ 2 Ausgänge

besteht aus

```
void comp(chan in1, in2, out1 out2){  
  a = <- in1  
  b = <- in2  
  
  if (a < b)  
    out1 <- a  
    out2 <- b  
  return void  
  
  out1 <- b  
  out2 <- a  
  return void  
}
```

Konvention für die folgenden Folien

- ▶ Reihenfolge des Inputs egal
- ▶ kleineres Element am oberen Ausgang
- ▶ größeres Element am unteren Ausgang

soll

- ▶ einfach
- ▶ anschaulich
- ▶ aber korrekt

zeigen dass etwas sortiert wurde

man kann jede Zahlenfolge durch eine 0,1 Folge repräsentieren

0,1- Beispiel

Motivation

Vorraussetzungen

Sortiernetzwerk

Aufbau

Sortieren im Sortiernetzwerk

Laufzeit

Fazit

Zusammenfassung

Ausblick

besteht aus mehreren Komperatoren

sort in sortnet

- ▶ Aufgabe
- ▶ grundlegendes Prinzip
- ▶ Demonstration (kleines Beispiel)
- ▶ Veranschaulichung an einem 2^x Beispiel
- ▶ zeigen dass Aufgabe erfüllt wird

- ▶ Aufgabe
- ▶ grundlegendes Prinzip
- ▶ Demonstration (kleines Beispiel)
- ▶ Veranschaulichung an einem 2^x Beispiel
- ▶ zeigen dass Aufgabe erfüllt wird

Motivation

Vorraussetzungen

Sortiernetzwerk

Laufzeit

Herleitung

Vergleich mit Software sortieren

Fazit

Zusammenfassung

Ausblick

Herleitung

Vergleich

Motivation

Vorraussetzungen

Sortiernetzwerk

Laufzeit

Fazit

Geschwindigkeit vs Variabilität

Hardwareaufwand vs Softwareaufwand

Zusammenfassung

Ausblick

Fazit

speed vs vari

hardware vs software

Motivation

Vorraussetzungen

Sortiernetzwerk

Laufzeit

Fazit

Zusammenfassung

Ausblick

zusammenfassung

Motivation

Vorraussetzungen

Sortiernetzwerk

Laufzeit

Fazit

Zusammenfassung

Ausblick

Hypercube

Anhang

weiter

?i

structur

Fragen, Anregungen?
(keine Liederwünsche)



A. Author.

Taschenbuch der Algorithmen.
Springer Verlag , 2008.



Tom Leighton.

Einführung in Parallele Algorithmen und Architekturen
Gitter, Bäume und Hypercubes.
Thomsom Publisching , 1997.

Make Titles Informative. Use Uppercase Letters. Long Titles are Split Automatically.

- ▶ Use `itemize` a lot.
- ▶ Kurze Sätze benutzen.

You can create overlays. . .

- ▶ using the pause command:
 - ▶ First item.

Make Titles Informative.

You can create overlays. . .

- ▶ using the pause command:
 - ▶ First item.
 - ▶ Second item.
- ▶ using overlay specifications:
- ▶ using the general uncover command:

Make Titles Informative.

You can create overlays. . .

- ▶ using the pause command:
 - ▶ First item.
 - ▶ Second item.
- ▶ using overlay specifications:
 - ▶ First item.
- ▶ using the general uncover command:

Make Titles Informative.

You can create overlays. . .

- ▶ using the pause command:
 - ▶ First item.
 - ▶ Second item.
- ▶ using overlay specifications:
 - ▶ First item.
 - ▶ Second item.
- ▶ using the general uncover command:

Make Titles Informative.

You can create overlays. . .

- ▶ using the pause command:
 - ▶ First item.
 - ▶ Second item.
- ▶ using overlay specifications:
 - ▶ First item.
 - ▶ Second item.
- ▶ using the general uncover command:
 - ▶ First item.

Make Titles Informative.

You can create overlays. . .

- ▶ using the pause command:
 - ▶ First item.
 - ▶ Second item.
- ▶ using overlay specifications:
 - ▶ First item.
 - ▶ Second item.
- ▶ using the general uncover command:
 - ▶ First item.
 - ▶ Second item.

```
int main (void)
{
    std::vector<bool> is_prime (100, true);
    for (int i = 2; i < 100; i++)
        if (is_prime[i])
        {
            std::cout << i << " ";
            for (int j = i; j < 100;
                is_prime [j] = false, j+=i);
        }
    return 0;
}
```

```
int main (void)
{
    std::vector<bool> is_prime (100, true);
    for (int i = 2; i < 100; i++)

        return 0;
}
```

```
int main (void)
{
    std::vector<bool> is_prime (100, true);
    for (int i = 2; i < 100; i++)
        if (is_prime[i])
        {

        }
    return 0;
}
```

```
int main (void)
{
    std::vector<bool> is_prime (100, true);
    for (int i = 2; i < 100; i++)
        if (is_prime[i])
        {
            std::cout << i << " ";
            for (int j = i; j < 100;
                is_prime [j] = false, j+=i);
        }
    return 0;
}
```

An Algorithm For Finding Primes Numbers.

```
int main (void)
{
    std::vector<bool> is_prime (100, true);
    for (int i = 2; i < 100; i++)
        if (is_prime[i])
        {
            std::cout << i << " ";
            for (int j = i; j < 100;
                is_prime [j] = false, j+=i);
        }
    return 0;
}
```

Note the use of `std::`.

Example

- ▶ 2 is prime (two divisors: 1 and 2).
- ▶ 3 is prime (two divisors: 1 and 3).
- ▶ 4 is not prime (**three** divisors: 1, 2, and 4).

Theorem

There is no largest prime number and, in addition,

$$\int_{\Omega} \nabla u \cdot \nabla v = - \int_{\Omega} u \Delta v + \int_{\partial \Omega} u v n$$

Proof.

1. Suppose p were the largest prime number.
- 2.
- 3.
4. Thus $q + 1$ is also prime and greater than p . □

Theorem

There is no largest prime number and, in addition,

$$\int_{\Omega} \nabla u \cdot \nabla v = - \int_{\Omega} u \Delta v + \int_{\partial \Omega} u v n$$

Proof.

1. Suppose p were the largest prime number.
2. Let q be the product of the first p numbers.
4. Thus $q + 1$ is also prime and greater than p . □

Theorem

There is no largest prime number and, in addition,

$$\int_{\Omega} \nabla u \cdot \nabla v = - \int_{\Omega} u \Delta v + \int_{\partial \Omega} u v n$$

Proof.

1. Suppose p were the largest prime number.
2. Let q be the product of the first p numbers.
3. Then $q + 1$ is not divisible by any of them.
4. Thus $q + 1$ is also prime and greater than p . □

Theorem

There is no largest prime number and, in addition,

$$\int_{\Omega} \nabla u \cdot \nabla v = - \int_{\Omega} u \Delta v + \int_{\partial \Omega} u v n$$

Proof.

1. Suppose p were the largest prime number.
2. Let q be the product of the first p numbers.
3. Then $q + 1$ is not divisible by any of them.
4. Thus $q + 1$ is also prime and greater than p . □

The proof used *reductio ad absurdum*.

Make Titles Informative.

- ▶ The **first main message** of your talk in one or two lines.
- ▶ The **second main message** of your talk in one or two lines.
- ▶ Perhaps a **third message**, but not more than that.

- ▶ Outlook
 - ▶ Something you haven't solved.
 - ▶ Something else you haven't solved.