



Parallele Sortierung

Björn Rathjen Patrick Winterstein Freie Universität Berlin

Proseminar Algorithmen, SS14



Outline I

Motivation

Allgemeir

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

Inhalt zusammesfassen

Ausblick

Hybercube

Anhang

Outline II

Previous Work

Our Results/Contribution
Main Results
Basic Ideas for Proofs/Implementation



Motivation Allgemein Bezug aufs Fach

Vorraussetzungen

Sortiernetzwerk

Laufzeit

Fazit

Inhalt zusammesfassen

Ausblick

mot

allgemein

fach



Vorraussetzungen Komparator 0,1-Prinzip

Sortiernetzwerk

Laufzeit

Fazit

Inhalt zusammesfassen

Ausblick



vorraussetzungen

Komparator

01 prinzip



Vorraussetzungen

Sortiernetzwerk Aufbau Sortieren im Sortiernetzwerk

Laufzeit

Fazit

Inhalt zusammesfassen

Ausblick

aufb sort

sort in sortnet

nativ

Beschreibung Teil 1



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

Beschreibung Teil 2



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



Vorraussetzungen

Sortiernetzwerk

Laufzeit Herleitung Vergleich mit Software sortieren

Fazit

Inhalt zusammesfassen

Ausblick



Vorraussetzungen

Sortiernetzwerk

Laufzeit

Fazit

Geschwindigkeit vs Variabilität Hardwareaufwand vs Softwareaufwand

Inhalt zusammesfassen

Ausblick



Vorraussetzunger

Sortiernetzwerk

Laufzeit

Fazit

Inhalt zusammesfassen

Ausblick

Vorraussetzungen

Sortiernetzwerk

Laufzeit

Fazit

Inhalt zusammesfasser

Ausblick Hybercube Anhang Previous Work

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



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



- using the pause command:
 - ► First item.



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



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



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



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



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



An Algorithm For Finding Primes Numbers.

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



Vorraussetzungen

Sortiernetzwerk

Laufzeit

Fazit

Inhalt zusammesfassen

Ausblick

Our Results/Contribution
Main Results
Basic Ideas for Proofs/Implementation



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).

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.

4. Thus q + 1 is also prime and greater than p.

7

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.

П



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.



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.



Summary



- ► 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.

For Further Reading I



A. Author.

Taschenbuch der Algorithmen. Springer Verlag, 2008.

Tom Leighton.

Einführung in Parallele Algorithmen und Architekturen Gitter, Bäume und Hypercubes.

Thomsom Publisching, 1997.