

Test test test

- Anzahl der Threads geregelt über von außen steuerbaren Parameter
- Graphen in unterschiedliche Klassen unterteilen
- Objekte werden in Oberklasse zusammengefasst
- MP oder SM ? noch zu klären
- Notiz: Max Wert von Initialisierung wird nie überschritten!
- Unterteilung der Graphstruktur in gröbere "Spalten", die aus mehreren Einzelspalten bestehen
- Jede Spalte wird von einem Thread bearbeitet

1 Allgemeines & Überblick

Zur Lösung des nebenläufigen Problems wird die Graphenstruktur insgesamt in m gröbere Spalten unterteilt, wobei m als steuerbarer Parameter implementiert werden kann. Zusätzlich werden sog. Synchronisationsgrenzen implementiert, die angeben, nach wie vielen Iterationsschritten die Akkumulatoren an den Spaltenübergängen synchronisiert werden. Dieser Parameter soll ebenfalls von außen steuerbar sein. Klassenstruktur: Ein Graph wird dynamisch aufgebaut. Dafür gibt es eine Klasse *klasse*. Ein Knoten entsteht, sobald sein Wert positiv wird. Die Knoten werden in einer Oberklasse *Oberklasse* zusammengesetzt. Zur spaltenweisen Abarbeitung wird es eine Klasse *Column?* geben

2 Nebenläufige Probleme

Nebenläufige Probleme treten vor allem bei der iterativen Berechnung der Spalten durch mehrere Threads, als auch der Synchronisation der Akkumulatoren auf. Dazu gehen wir folgenden Weg: