

Benutzerhandbuch

Es liegen hier 2 Ordner vor. Der Ordner Histogramme enthält jeweils alle py-Dateien, welche zum Plotten der Histogramme benötigt werden. Diese liegen für die konstante Lambda-Funktion, Lambda1 wie auch für die Lambda2-Funktion vor. Die Tests entsprechen denen aus der Bachelorarbeit.

Der zweite Ordner Unrelated-Scheduling, enthält alle C++-Files und alles was zusätzlich durch Visual-Studio generiert wurde. Das Programm wird in main.cpp ausgeführt. Anders als in der Bachelorarbeit ist A statt x die Zuweisungsmatrix, und T statt t ist die Eingabe. In den Matrizen befinden sich zusätzlich auf der Diagonalen nur Null-Einträge. Zudem ist jeder Index um Eins nach unten verschoben. Mittels float test_set[10] kann der Nutzer 10 Zahlen eingeben, die in der Eingabe vorkommen sollen. Jede Zahl aus dem Array wird mit einer Wahrscheinlichkeit von 0.1 vorkommen. Möchte man, dass eine Zahl häufiger vorkommt, so muss man entsprechend eine Zahl häufiger im Array setzen. Ist man mit dieser manuellen Setzung fertig so kann das Programm gestartet werden. Es wird empfohlen, es im Release-Modus auszuführen. Im Debug-Modus läuft das Programm wesentlich langsamer.

Der Nutzer wird aufgefordert zu entscheiden wie viele Spieler mitmachen dürfen. Hier darf der Nutzer nur eine 5 oder 6 angeben. Ansonsten wird dieser aufgefordert eine neue Zahl anzugeben.

Anschließend wird dieser dazu aufgefordert die Anzahl der Testfälle anzugeben. Hier ist jede Zahl > 0 erlaubt.

Als Drittes wird dieser dazu aufgefordert zu entscheiden, ob dieser die konstante, Lambda1, Lambda2, Lambda3 oder Lambda4-Funktion testen möchte. In der Arbeit haben wir zu den letzten beiden Funktionen keine Testfälle angegeben. Übersichtshalber habe ich diese hier dennoch hinzugefügt. Die Lambda5_k-Funktion macht für $n \leq 6$ wenig Sinn. Entsprechend kommt dieser hier auch nicht vor.

Entscheidet sich der Nutzer für die konstante Lambda-Funktion so wird dieser aufgefordert alle Lambda_{ij} Werte mit $j > i$ anzugeben. Lambda_{ij} > 2 und Lambda_{ij} ≤ 0 sind nicht erlaubt. Ansonsten muss der Nutzer die entsprechenden Werte erneut angeben.

Anschließend wird das Programm normal gestartet.

Bei der Lambda1-, Lambda2-, Lambda3- und Lambda4-Funktion wird der Nutzer zusätzlich dazu aufgefordert ein Lambda_max anzugeben. Dieses muss ebenso zwischen 1 und 2 liegen.

Je nach Anzahl der Testfälle (für $n=5$ für 10 Millionen ca 10 Minuten, für $n=6$ bei einer Millionen ca 30 Minuten) wird das Programm zunächst Berechnungen durchführen.

Am Ende erhalten wir folgende Ausgabe:

Für die Funktion werden jeweils in 0.1 Schritten angegeben wie groß der errechnete Approximationsfaktor einer Instanz ist. Fällt ein Wert bspw auf die 3.8 fällt, so wissen wir, dass der eigentliche Approximationsfaktor im Intervall $(3.7, 3.8]$ liegt.

Weiter wird der jeweils größte und auch der jeweils kleinste wie auch der durchschnittliche Approximationsfaktor angegeben. Zudem wird angegeben wie viele Instanzen einen Approximationsfaktor von mindestens $n-1$ erreichen konnten.

Zudem werden jeweils noch 5 Worst-Case-Fälle angegeben. Bei der Lambda1-Funktion wird zusätzlich jede errechnete Matrix von OPT_{ij} auch angegeben.