Abteilung für
Angewandte Mathematik

SS 2024
Ludwig Striet

Einführung in die Programmierung für Studierende der Naturwissenschaften

Blatt 9 – 24.06.2024

Abgabe: *Bis 30.06.2024, 23:59 Uhr per E-Mail an Ihre/n Tutor/in*

Aufgabe 1 (5 Punkte)**.** Verwenden Sie OpenMP um eine Funktion

```
1       int parallel_min(int* L, unsigned int n, unsigned int np)
2       {
3         // ...
4       }
```

zu schreiben, die `nt` Threads benutzt, um das Minimum in einer Liste `L` der Länge `n` zu finden.

Aufgabe 2 (5 Punkte)**.** Auf der Vorlesungswebseite finden Sie die Datei `sort_compare.cc` in der das Mergesort Verfahren so implementiert ist, dass zwei threads gleichzeitig verwendet werden um eine Liste zu sortieren. Ändern Sie die Implementierung so, dass ein zusätzliche Parameter `unsigned int nt` übergeben wird und dass `nt` viele Threads verwendet werden, um die Liste zu sortieren.

Aufgabe 3 ($3 + 3 + 4$ Punkte)**.** Mit Hilfe einer `std::priority_queue` kann eine Liste der Länge N sortiert werden, in dem erst alle Elemente nacheinander mit `push` in die queue eingefügt werden und dann mit `top` einzeln ausgelesen und mit `pop` wieder aus der `std::priority_queue` gelöscht werden.

(a) Erläutern Sie, warum das Verfahren die Daten sortiert

(b) Das Einfügen eines Elements in eine `std::priority_queue` der Länge N mit Hilfe des `push`-Befehls hat Kosten $C \cdot \log(N)$ und das entfernen eines Elements mit Hilfe des `pop`-Befehls hat ebenfalls Kosten $C \cdot \log(N)$. Begründen Sie, warum das SortierenVerfahren aus (a) Aufwand $C \cdot N \log(N)$ für eine Konstante C hat.

(c) Implementieren Sie das Verfahren unter Verwendung der `std::priority_queue`. Es ist ausreichend, wenn Sie das Verfahren für Daten vom Typ `int` implementieren. Die Daten dürfen auch in absteigender Reihenfolge sortiert werden. Die Daten müssen nicht sortiert wieder gespeichert werden, es reicht, wenn sie sortiert ausgegeben werden.

Hinweis: Es ist Teil der Aufgabe, dass Sie sich mit der Funktionsweise von `std::priority_queue` vertraut machen. Eine gute Anlaufstelle ist die Referenz unter `www.cppreference.com`.