

Task 3

Orientiert man sich an den in Aufgabe 2 vorgegebenen Schritten 1-6 und ein wenig am selbst geschriebenen Code, so müssen folgenden Laufzeiten summiert werden:

| | | | | | | | |
|--|---|---|---|---|---|---|---|
| $n \cdot \log(n)$ | + | n | + | $\log(p + (p-1) \cdot \text{size}(\text{int}))$ | + | $(p \cdot (p-1)) \cdot \log(p \cdot (p-1))$ | + |
| Quicksort | | Splitter in lokalen Arrays identifizieren | | MPI_Gather | | Quicksort im root Prozess auf gesammelten Splittern | |
| $p \cdot (p-1)$ | + | $(p-1) \cdot \text{size}(\text{int}) \cdot \log(p)$ | + | n | + | | |
| Globale Splitter Identifizieren | | MPI_Bcast | | Lokale Splitter auf Grundlage von globalen Splittern identifizieren | | | |
| $p + (p-1) \cdot \text{size}(\text{int})$ | + | $p + n$ | + | $n \cdot \log(n)$ | | | |
| MPI_Alltoall Versende die zu erwartenden Größe eines Blockes an alle Blöcke | | MPI_Alltoallv Versende Blöcke an alle Prozesse | | Quicksort auf den empfangenen zusammengeführten Blöcken | | | |

Mit der Annahme, dass n sehr viel größer als p ist, vereinfacht sich die Summe zu:

$$3n + 2n \cdot \log(n)$$

Dies bedeutet das die Laufzeit unseres Algorithmus in der **Komplexitätsklasse $O(n \cdot \log(n))$** liegt.

Der Speicherbedarf im root Prozess ist, auf Grund der Sonderfunktion, am höchsten und wird daher hier betrachtet. Wenn man sich an der Aufgabenstellung zwei orientiert, beläuft er sich auf

$$n \cdot \text{sizeof}(\text{int}) + 1 \cdot \text{sizeof}(\text{int}) + (p-1) \cdot \text{sizeof}(\text{int}) + p \cdot (p-1) \cdot \text{sizeof}(\text{int}) + n \cdot \text{sizeof}(\text{int}) \\ = (2n + p^2) \cdot \text{sizeof}(\text{int})$$

Die einzelnen Summanden stellen folgenden Bedarf da:

Das lokale Array + die Länge des lokalen Arrays + die lokalen Splitter + die Summe der Splitter, die im root-Prozess gesammelt werden + ein weiteres Array in dem die empfangenen Blöcke zusammengeführt werden.

Die **Obergrenze für unseren Speicher**, bei Annahme $n \gg p$, liegt also bei **$O(n)$** .