## Benchmarking Cache Efficiency

# Bubble Sort

Theoretische Komplexität (Time/Space):

Beispiel eines Best-Case Szenarios:

Beispiel eines Worst Case Szenarios:

Mögliche Einschränkungen bei der Anwendbarkeit:

# Insertion Sort

Theoretische Komplexität (Time/Space):

Beispiel eines Best-Case Szenarios:

Beispiel eines Worst Case Szenarios:

Mögliche Einschränkungen bei der Anwendbarkeit:

# Quick Sort

Theoretische Komplexität (Time/Space):

Beispiel eines Best-Case Szenarios:

Beispiel eines Worst Case Szenarios:

Mögliche Einschränkungen bei der Anwendbarkeit:

# Bucket Sort

Theoretische Komplexität (Time/Space):

Beispiel eines Best-Case Szenarios:

Beispiel eines Worst Case Szenarios:

Mögliche Einschränkungen bei der Anwendbarkeit:

# Gewählte Variationen

1. **Variante 1 (Basisvariante)**
   1. Gesamtes Datenarray passt in den CPU-Cache
   2. Array Eintrag ist ein simpler Integer
   3. Array enthält direkt die konkreten Inhalte (Array of Stucts/Primitives)
2. **Variante 2**
   1. **Array ist zu groß für den Cache**
   2. Array Eintrag ist ein simpler Integer
   3. Array enthält direkt die konkreten Inhalte (Array of Stucts/Primitives)
3. **Variante 3**
   1. Gesamtes Datenarray passt in den CPU-Cache
   2. **Array Eintrag ist ein komplexes Struct mit mehreren Variablen (und einem Integer Key)**
   3. Array enthält direkt die konkreten Inhalte (Array of Stucts/Primitives)
4. **Variante 4**
   1. Gesamtes Datenarray passt in den CPU-Cache
   2. Array Eintrag ist ein simpler Integer
   3. **Jeder einzelne Inhalt wird separat erzeugt(allokiert). Das Array enthält nur Referenzen oder Pointer auf diese Inhalte.**
5. **Variante 5**
   1. **Array ist zu groß für den Cache**
   2. **Array Eintrag ist ein komplexes Struct mit mehreren Variablen (und einem Integer Key)**
   3. Array enthält direkt die konkreten Inhalte (Array of Stucts/Primitives)
6. **Variante 6**
   1. **Array ist zu groß für den Cache**
   2. Array Eintrag ist ein simpler Integer
   3. **Jeder einzelne Inhalt wird separat erzeugt(allokiert). Das Array enthält nur Referenzen oder Pointer auf diese Inhalte.**
7. **Variante 7**
   1. Gesamtes Datenarray passt in den CPU-Cache
   2. **Array Eintrag ist ein komplexes Struct mit mehreren Variablen (und einem Integer Key)**
   3. **Jeder einzelne Inhalt wird separat erzeugt(allokiert). Das Array enthält nur Referenzen oder Pointer auf diese Inhalte.**
8. **Variante 8**
   1. **Array ist zu groß für den Cache**
   2. **Array Eintrag ist ein komplexes Struct mit mehreren Variablen (und einem Integer Key)**
   3. **Jeder einzelne Inhalt wird separat erzeugt(allokiert). Das Array enthält nur Referenzen oder Pointer auf diese Inhalte.**

Alle Variationen wurden in der Programmiersprache C++ implementiert. Fett gedruckt sind jeweils die Unterschiede zur Basisvariante.

# Benchmark Maschine

# Verwendete Compiler und Compilereinstellungen

# Gemessene Laufzeiten im Vergleich

# Interpretation der Ergebnisse