

AUFGABEN (DEUTSCH)

Aufgabe 1 (3 PUNKTE): Implementieren Sie eine rekursive und eine nicht rekursive Version des insertion sort Algorithmus und testen Sie beide mit einem Array von Integerwerten. Füllen Sie Ihr Array mit einer großen Anzahl von Elementen (> 1000) und berechnen Sie die Zeit, die beide Algorithmen benötigen, um das Array zu sortieren (führen Sie den Code mehrmals aus und berechnen Sie einen Durchschnitt der Zeit über die mehreren Durchläufe). Reichen Sie den C++-Code (cpp-Dateien, h-Dateien und auch ein Makefile) für diese Anwendung zusammen mit einem Bildschirmfoto des Eingabe-Arrays und des Ausgabe-Arrays (das sortiert sein sollte!) ein, das Sie beim Ausführen des Codes mit einem Array von nur 50 Elementen erhalten haben.

Aufgabe 2 (3 PUNKTE): Implementieren Sie den selection sort Algorithmus und testen Sie diesen mit einem Array mit ganzzahligen Werten. Füllen Sie Ihr Array mit einer großen Anzahl von Elementen (> 1000) und berechnen Sie die Zeit, die der Algorithmus benötigt, um ein bestimmtes Element zu finden (führen Sie den Code mehrmals aus und berechnen Sie den Durchschnitt der Zeit über die verschiedenen Durchläufe). Reichen Sie den C++-Code (cpp-Dateien, h-Dateien und auch ein Makefile) für diese Anwendung zusammen mit einem Bildschirmfoto des Eingabe-Arrays und des Ausgabe-Arrays (das sortiert sein sollte!) ein, das Sie beim Ausführen des Codes mit einem Array von nur 50 Elementen erhalten haben.

TASKS (ENGLISH TEXT)

Task 1 (3 POINTS): Implement a recursive and a non-recursive version of the insertion sort and test both of them with an array of integer values. Populate your array with a large number of elements (> 1000) and calculate the time it takes to both algorithms to sort the array (run the code multiple times, and calculate an average of the time over the multiple running sessions). Submit the C++ code (cpp files, h files, and also a makefile) for this application along with a screen shot of the input array and the output array (which should be sorted!) obtained running the code with an array of 50 elements only.

Task 2 (3 POINTS): Implement the selection sort algorithm and test it on an array of integer values. Populate your array with a large number of elements (> 1000) and calculate the time it takes the algorithm to find a given element (run the code multiple times, and calculate an average of the time over the multiple running sessions). Submit the C++ code (cpp files, h files, and also a makefile) for this application along with a screen shot of the input array and the output array (which should be sorted!) obtained running the code with an array of 50 elements only.