

Bubblesort (Blasensortierung)

Die Grundidee ist, dass die Liste immer wieder durchlaufen wird und dabei benachbarte Elemente in die richtige Reihenfolge gebracht werden. Größere Elemente überholen so die kleineren und drängen sich an das Ende der Liste („wie aufsteigende (Luft-)Blasen“).

Beschreibung des Algorithmus:

Die Liste wird von links nach rechts durchlaufen. Dabei wird bei jedem Schritt das aktuelle Element mit seinem rechten Nachbarn verglichen:

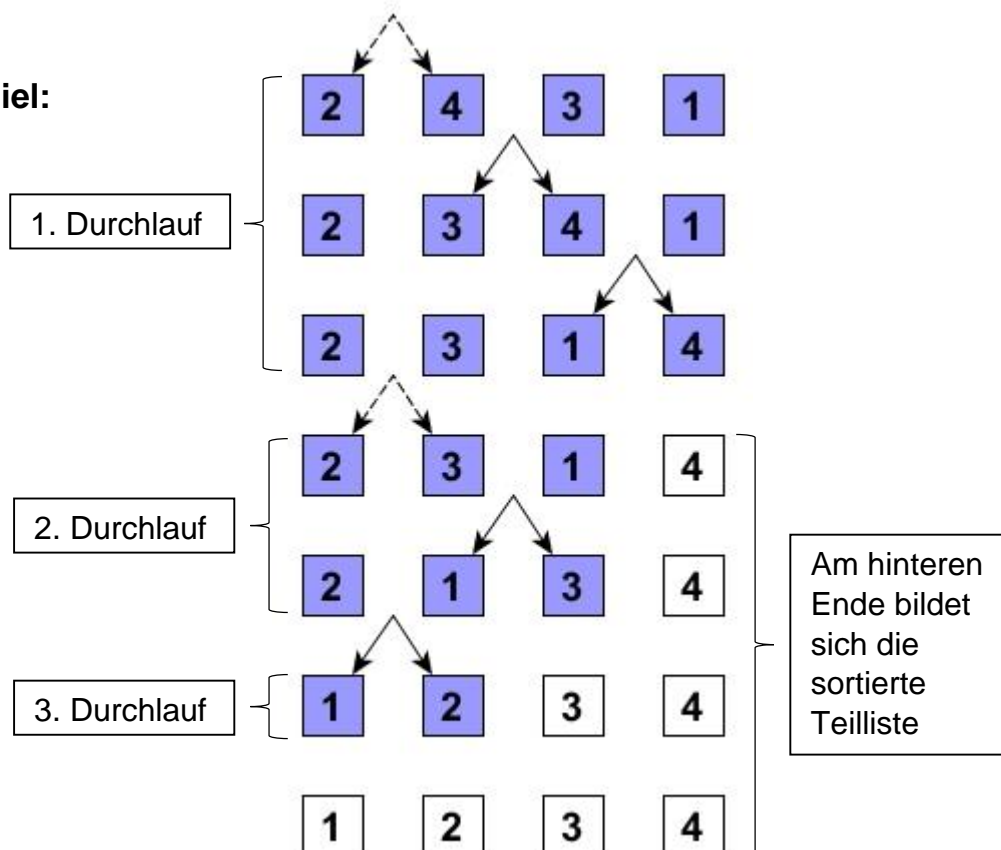
- Falls das aktuelle Element größer (schwerer) ist als sein rechter Nachbar werden die beiden Elemente getauscht.
- Falls der rechte Nachbar größer (schwerer) ist als das aktuelle Element soll nichts gemacht werden.

Nun werden die beiden nächsten beiden Elemente miteinander verglichen.

Nach dem ersten Durchlauf ist das größte (schwerste) Element am Ende der Liste angekommen und ist damit sortiert. Somit muss dieses Element beim zweiten Durchlauf nicht mehr betrachtet werden.

Die Liste ist sortiert, wenn bei einem Durchlauf keine Elemente mehr getauscht werden müssen.

Beispiel:



Aufgaben

1.) Lese die Anleitung samt Beispiel zu dem gegebenen Algorithmus genau durch. Hole einen Materialiensatz vom Lehrerpult.

2.) Sortiere die Cornflakes-Packungen mithilfe des gegebenen Algorithmus aufsteigend nach Gewicht.

Die Startreihenfolge dafür ist: Cornflakes 1, Honey Bss Pops, Cornflakes 2, Choco Krispies Chocos, Frosties, Smacks, Choco Krispies, Special K Classic.



Notiere zudem, wie häufig mit diesem Algorithmus gewogen werden muss, bis die Cornflakes-Packungen nach Gewicht sortiert sind.

3.) Schätze anhand der Anzahl der zu vergleichenden Elemente im Vergleich zu der Anzahl der Wiegungen ab, in welcher Laufzeitkategorie der Algorithmus liegen müsste. (Bedenke, dass der Computer für jede Wiegung mehr als einen Rechenschritt benötigt!)

4.) Begründe in Worten anhand der Beschreibung des Algorithmus, warum deine Abschätzung der Laufzeitkategorie stimmt.

Zusatzaufgaben:

Z1) Begründe, warum es zulässig ist, nur die Anzahl der Wiegungen zur Abschätzung des Laufzeitaufwands heranzuziehen.

Z2) Überlege, wie die Startreihenfolge angeordnet sein muss, so dass der Algorithmus nach möglichst wenigen Schritten beendet ist.

Bestimme, in welcher Laufzeitkategorie der Algorithmus mit dieser (speziellen) Startreihenfolge liegen würde.