

CE2001/ CZ2001: Algorithms

Introduction to Sorting

Ke Yiping, Kelly

Instructor's Information

- **Name:** Ke Yiping, Kelly
- **Phone:** 6790-5046
- **Email:** ypke@ntu.edu.sg
- **Office:** N4-02a-12

2

Learning Objectives

At the end of this lecture, students should be able to:

- Define what is sorting
- Explain why we learn sorting
- Analyse the objective and evaluation of sorting algorithms

3

What is Sorting?

Definition (sorting in ascending order):

- Given a set of records r_1, r_2, \dots, r_n with key values k_1, k_2, \dots, k_n , arrange records in order s such that records $r_{s1}, r_{s2}, \dots, r_{sn}$ have keys with property $k_{s1} \leq k_{s2} \leq \dots \leq k_{sn}$.

Example:

Given	8	0	4	9	3	6
Sorted						



4

What is Sorting?

Definition (sorting in ascending order):

- Given a set of records r_1, r_2, \dots, r_n with key values k_1, k_2, \dots, k_n , arrange records in order s such that records $r_{s1}, r_{s2}, \dots, r_{sn}$ have keys with property $k_{s1} \leq k_{s2} \leq \dots \leq k_{sn}$.

Example:

Given	8	0	4	9	3	6
Sorted	0					



5

What is Sorting?

Definition (sorting in ascending order):

- Given a set of records r_1, r_2, \dots, r_n with key values k_1, k_2, \dots, k_n , arrange records in order s such that records $r_{s1}, r_{s2}, \dots, r_{sn}$ have keys with property $k_{s1} \leq k_{s2} \leq \dots \leq k_{sn}$.

Example:

Given	8	0	4	9	3	6
Sorted	0	3				



6

What is Sorting?

Definition (sorting in ascending order):

- Given a set of records r_1, r_2, \dots, r_n with key values k_1, k_2, \dots, k_n , arrange records in order s such that records $r_{s1}, r_{s2}, \dots, r_{sn}$ have keys with property $k_{s1} \leq k_{s2} \leq \dots \leq k_{sn}$.

Example:

Given	8	0	4	9	3	6
Sorted	0	3	4			



7

What is Sorting?

Definition (sorting in ascending order):

- Given a set of records r_1, r_2, \dots, r_n with key values k_1, k_2, \dots, k_n , arrange records in order s such that records $r_{s1}, r_{s2}, \dots, r_{sn}$ have keys with property $k_{s1} \leq k_{s2} \leq \dots \leq k_{sn}$.

Example:

Given	8	0	4	9	3	6
Sorted	0	3	4	6		



8

What is Sorting?

Definition (sorting in ascending order):

- Given a set of records r_1, r_2, \dots, r_n with key values k_1, k_2, \dots, k_n , arrange records in order s such that records $r_{s1}, r_{s2}, \dots, r_{sn}$ have keys with property $k_{s1} \leq k_{s2} \leq \dots \leq k_{sn}$.

Example:

Given	8	0	4	9	3	6
			↓			
Sorted	0	3	4	6	8	

9

What is Sorting?

Definition (sorting in ascending order):

- Given a set of records r_1, r_2, \dots, r_n with key values k_1, k_2, \dots, k_n , arrange records in order s such that records $r_{s1}, r_{s2}, \dots, r_{sn}$ have keys with property $k_{s1} \leq k_{s2} \leq \dots \leq k_{sn}$.

Example:

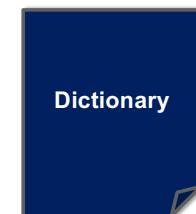
Given	8	0	4	9	3	6
			↓			
Sorted	0	3	4	6	8	9

10

Why do we learn sorting?

Why do we learn sorting?

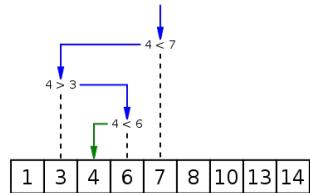
- Things must be kept in some order if we want to find them quickly.



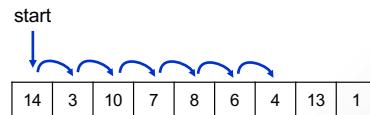
Why do we learn sorting?

- Things must be kept in some order if we want to find them quickly.
- How to arrange things in order? Sorting algorithms.
- Sorting is a basic building block for many algorithms.

Binary Search



Sequential Search



Reference: T. (2015, April 19). Binary search in a sorted array. Retrieved May 18, 2016, from https://commons.wikimedia.org/wiki/File:Binary_search_into_array.png#/media/File:Binary_search_into_array.png.

13

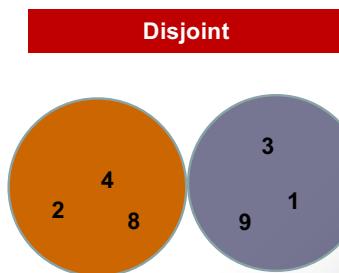
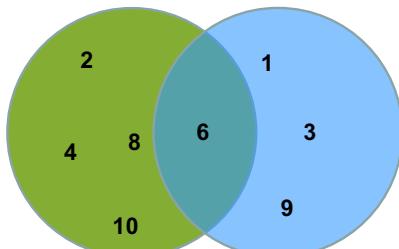
Why do we learn sorting?

- Things must be kept in some order if we want to find them quickly.
- How to arrange things in order? Sorting algorithms.
- Sorting is a basic building block for many algorithms.
- Most thoroughly studied problem in Computer Science.
- To learn ideas in Algorithm Design derived from techniques in sorting.

14

Example: Disjoint Sets

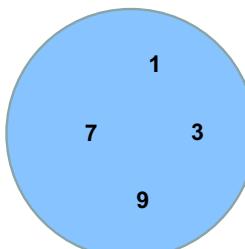
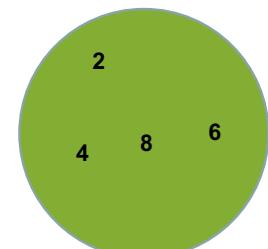
- **Problem:** Determine whether two sets (both of size n) are disjoint.



15

Example: Disjoint Sets

- **Problem:** Determine whether two sets (both of size n) are disjoint.
 - **Solution 1:** Compare each element of the 1st set with each element of the 2nd set. That is, n^2 comparisons.



16

Example: Disjoint Sets

- **Problem:** Determine whether two sets (both of size n) are disjoint.
- **Solution 1:** Compare each element of the 1st set with each element of the 2nd set. That is, n^2 comparisons.
- **Solution 2:**
 - Step 1:** We first **sort the first set into ascending order**. This takes $O(nlgn)$ effort using Mergesort or Heapsort.
 - Step 2:** For each element in the 2nd set, we **use Binary Search** to find it in the 1st set. This takes $O(nlgn)$ time.

17

Comparison of Performance

Solution 1: $O(n^2)$

Solution 2: $O(nlgn)$

Savings:

n	=	64	128	256	512
n^2	=	4,096	16,384	65,536	262,144
$nlgn$	=	384	896	2,048	4,608

18

Comparison of Performance

The data items to be sorted:

- Given a (very large) list of records.
- Each record has the following form: key; rest info of record:

```
class ALIST {
    KeyType   key;
    DataType  data;
};
```
- Key domain is an ordered set.
- **Objective:** To arrange records in 'ascending' or 'descending' order.

19

Comparison of Performance

The data items to be sorted:

- Given a (very large) list of records.
- Each record has the following form: key; rest of record:

```
class ALIST {
    KeyType   key;
    DataType  data;
};
```
- Key domain is an ordered set.
- **Objective:** To arrange records in 'ascending' or 'descending' order.

20

Comparison of Performance

- Sorting can be classified into **internal sorting** and **external sorting**.
 - We focus on internal sorting only,
i.e., all records are in (high speed) main memory during sorting.
- **Sorting involves two basic actions:**
 - 1) key comparisons between two records
 - 2) swapping records around
- **Goal:** Use **minimum** working space and do as **few** key comparisons as possible.

21

Summary

- Sorting is to arrange a set of records so that their key values are in ascending or descending order.
- It is important to learn sorting, because:
 - Sorting has important applications
 - Ideas of sorting can be used for other algorithms
- Objective is to design sorting algorithms with:
 - Minimum usage of memory
 - Minimum number of key comparisons or swaps

22