CH-231-A

Algorithms and Data Structures

ADS

Lecture 6

Dr. Kinga Lipskoch

Spring 2022

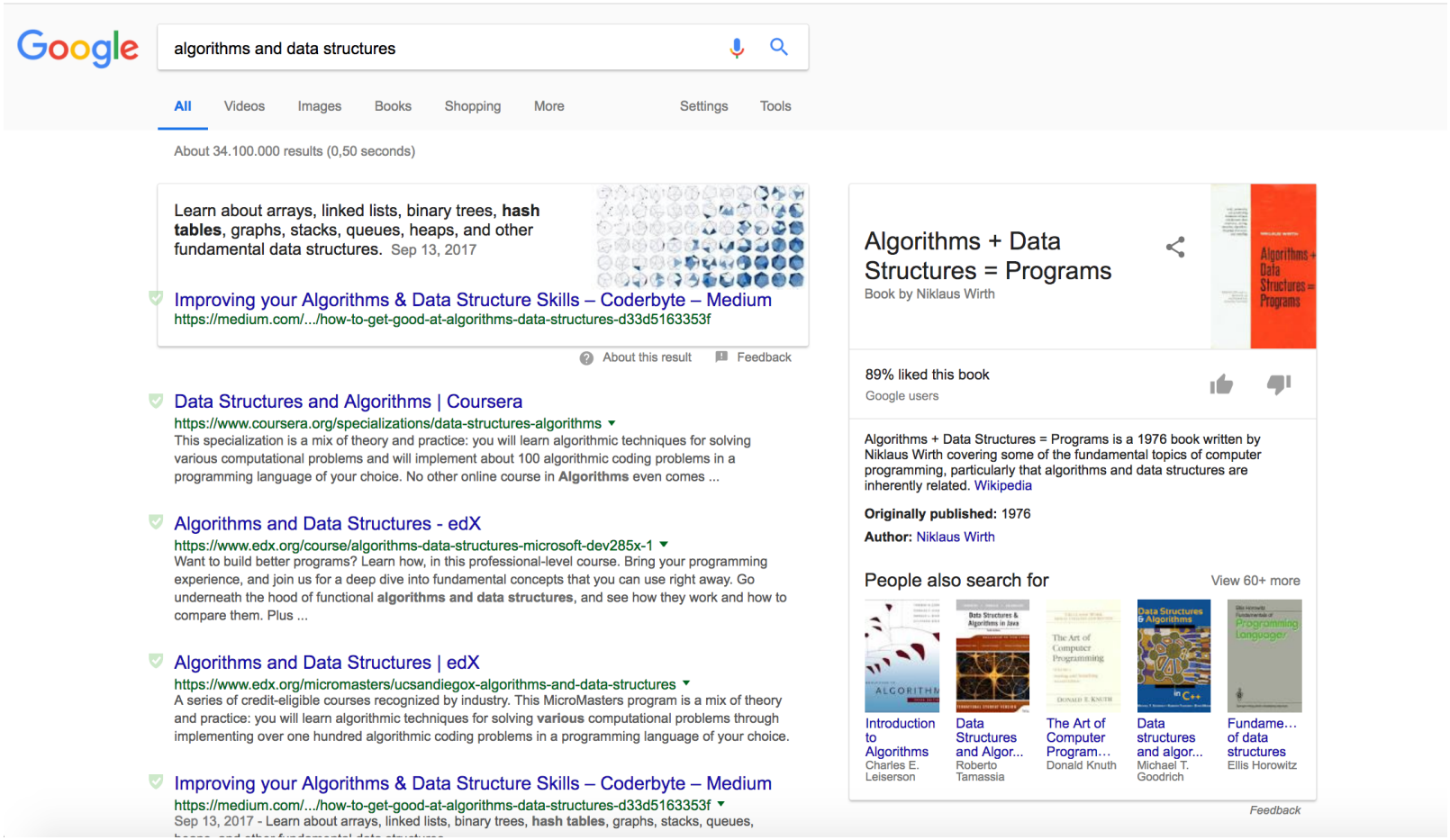
Definition: Algorithm

- An algorithm is a sequence of computational steps which transforms a set of values (input) to another set of values (desired output)
- It is a tool for solving a well-defined computational problem
- Step-wise procedure that can be implemented in a computer program
- Consists of a finite list of well-defined instructions (Turing machine)
- 'Algorithm' stems from 'Algoritmi', the Latin form of al-Khwārizmī, a Persian mathematician, astronomer and geographer

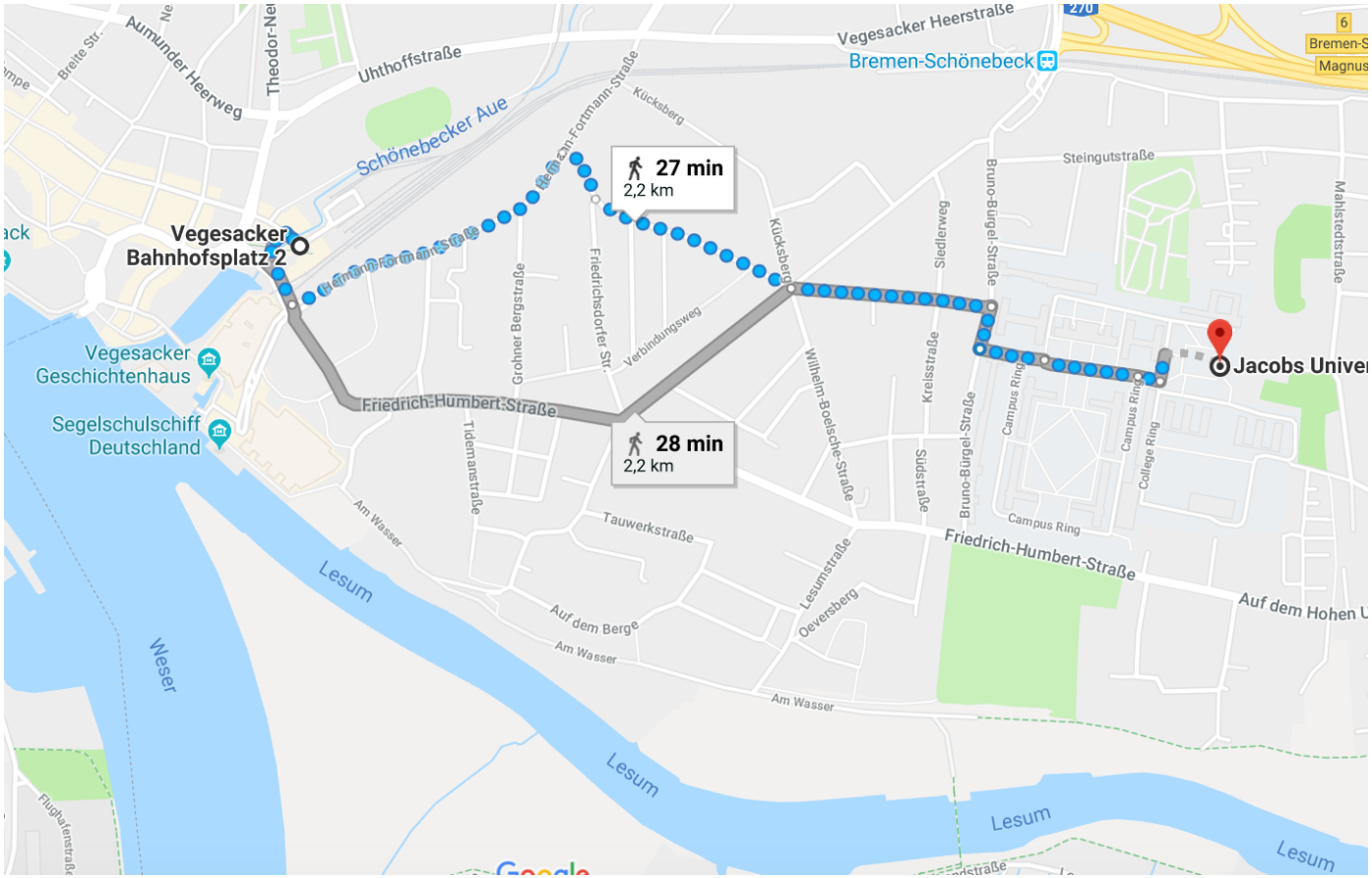
Example: Sorting Problem

- Input
  sequence $< a_1, a_2, ..., a_n >$ of numbers
- Output
  permutation $< a'_1, a'_2, ..., a'_n >$
  such that $a'_1 \leq a'_2 \leq ... \leq a'_n$
- Example (instance of sorting problem):
  Input: 8 2 4 9 3 6
  Output: 2 3 4 6 8 9

Example: Searching

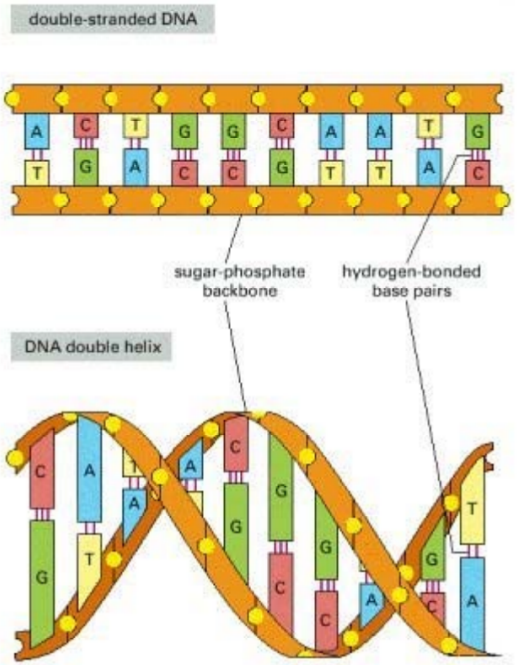


Example: Road map



Graph algorithm

Example: DNA Sequences



String matching

Analysis of Algorithms

- The theoretical study of computer-program performance and resource usage
- Other design goals?
  - correctness
  - functionality
  - robustness
  - reliability
  - user-friendliness
  - programmer time
  - simplicity
  - modularity
  - maintainability
  - extensibility

Performance of Algorithms

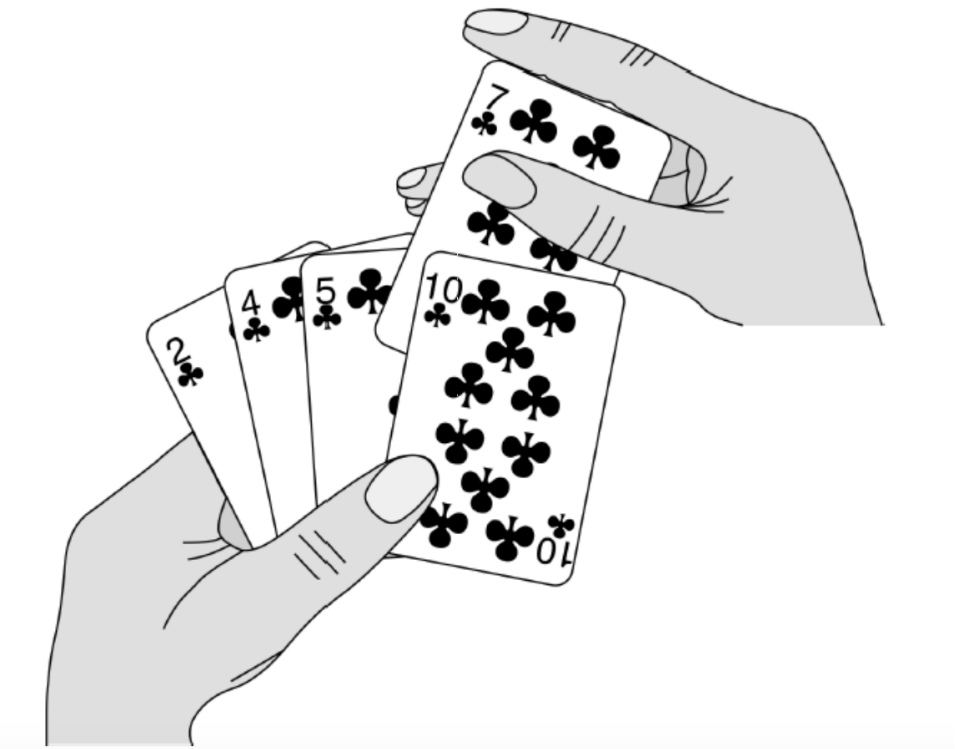
- Analysis helps us to understand scalability
- Performance often draws the line between what is feasible and what is impossible
- Algorithmic mathematics provides a language for talking about program behavior
- "Performance is the currency of computing"
- The lessons of program performance generalize to other computing resources

Definition: Data Structure

- A data structure is a way to store and organize data in order to facilitate access and modification
- There is typically no best data structure, but each data structure has its strengths and weaknesses
- Which data structure to use, depends on the problem that is to be solved
- Sometimes there is a trade-off between storage (in a data structure) and speed (in accessing a data structure or of an algorithm)

Sorting Problem

First algorithm: Insertion sort

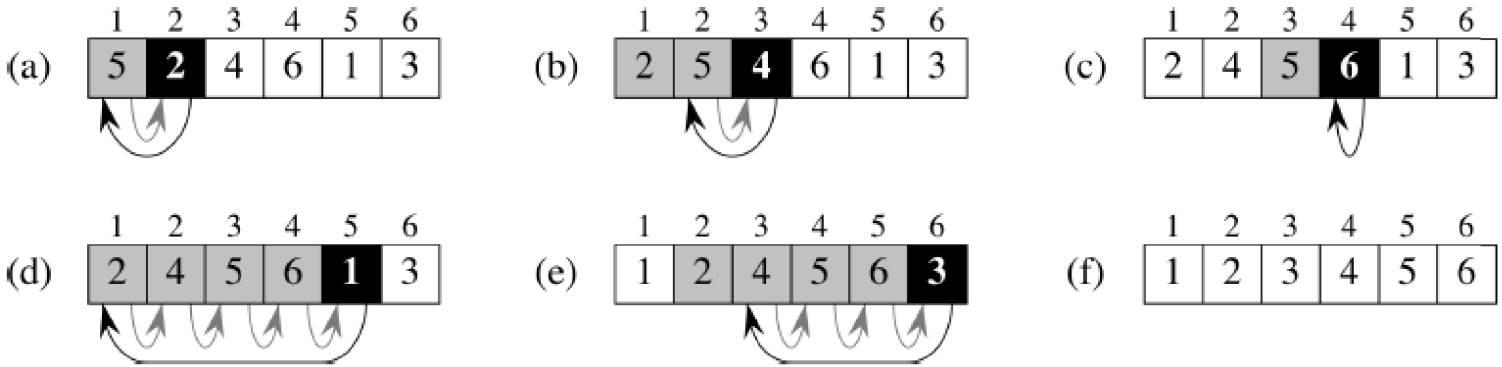


Insertion Sort

```
INSERTION-SORT(A, n)
  for j = 2 to n
      key = A[j]
      // Insert A[j] into the sorted sequence A[1 .. j − 1].
      i = j − 1
      while i > 0 and A[i] > key
          A[i + 1] = A[i]
          i = i − 1
      A[i + 1] = key
```

Insertion Sort: Example

Sort $A =< 5, 2, 4, 6, 1, 3 >$



INSERTION-SORT(A, n)

for $j = 2$ **to** n
  $key = A[j]$
  // Insert $A[j]$ into the sorted sequence $A[1 \,..\, j-1]$.
  $i = j - 1$
  while $i > 0$ and $A[i] > key$
    $A[i+1] = A[i]$
    $i = i - 1$
  $A[i+1] = key$