

Using Advanced Data Structures in Modern Applications

INTRODUCTION



Rasmus Resen Amossen

SOLUTION ARCHITECT

rasmus.resen.org

What (Not) to Expect

010010

101001

010111

Container

Lemma 3.11 Let $j \in \mathcal{B}$ satisfy $|A_i||C_i| \leq |A_j||C_j|$ for all $i \in \mathcal{B}$. Then $p = \min(1/k, k/(|A_j||C_j|))$ gives an expected $O(n)$ running time for Algorithm 1.

PROOF We argue that for each i , $p|A_i||C_i| \leq \max(|A_i|, |C_i|)$, which by Lemma 3.10 implies running time

$$O(n + \sum_i p|A_i||C_i|) = O(n + \sum_i \max(|A_i|, |C_i|)) = O(n).$$

Suppose first that $|A_i||C_i| \geq k^2$. Then $p = k/(|A_j||C_j|)$ and $p|A_i||C_i| \leq \sqrt{|A_i||C_i|} \leq \max(|A_i|, |C_i|)$. Otherwise, when $|A_i||C_i| < k^2$, we have $p|A_i||C_i| = |A_i||C_i|/k \leq \max(|A_i|, |C_i|)$.



```
5
6
7 class InvestmentAnalyzer
8 {
9     IStockTrader stockTrader;
10     CS.IntervalHeap<InvestmentQuery> queries = new CS.IntervalHeap<InvestmentQuery>();
11     Dictionary<string, int> stockRating = new Dictionary<string, int>();
12     HashSet<Guid> blackListedInvestors = new HashSet<Guid>();
13     Random random = new Random(23);
14
15     public InvestmentAnalyzer(IStockTrader stockTrader)
16     {
17         this.stockTrader = stockTrader;
18     }
19
20     public void HandleQuery(InvestmentQuery query)
21     {
22         queries.Add(query);
23     }
24
25     public void AnalyzeQueries()
26     {
27         while(!queries.IsEmpty)
28         {
29             int rating;
```

Prerequisites

Complexity analysis

Big-O notation

$O(\dots)$ $\Theta(\dots)$ $\Omega(\dots)$

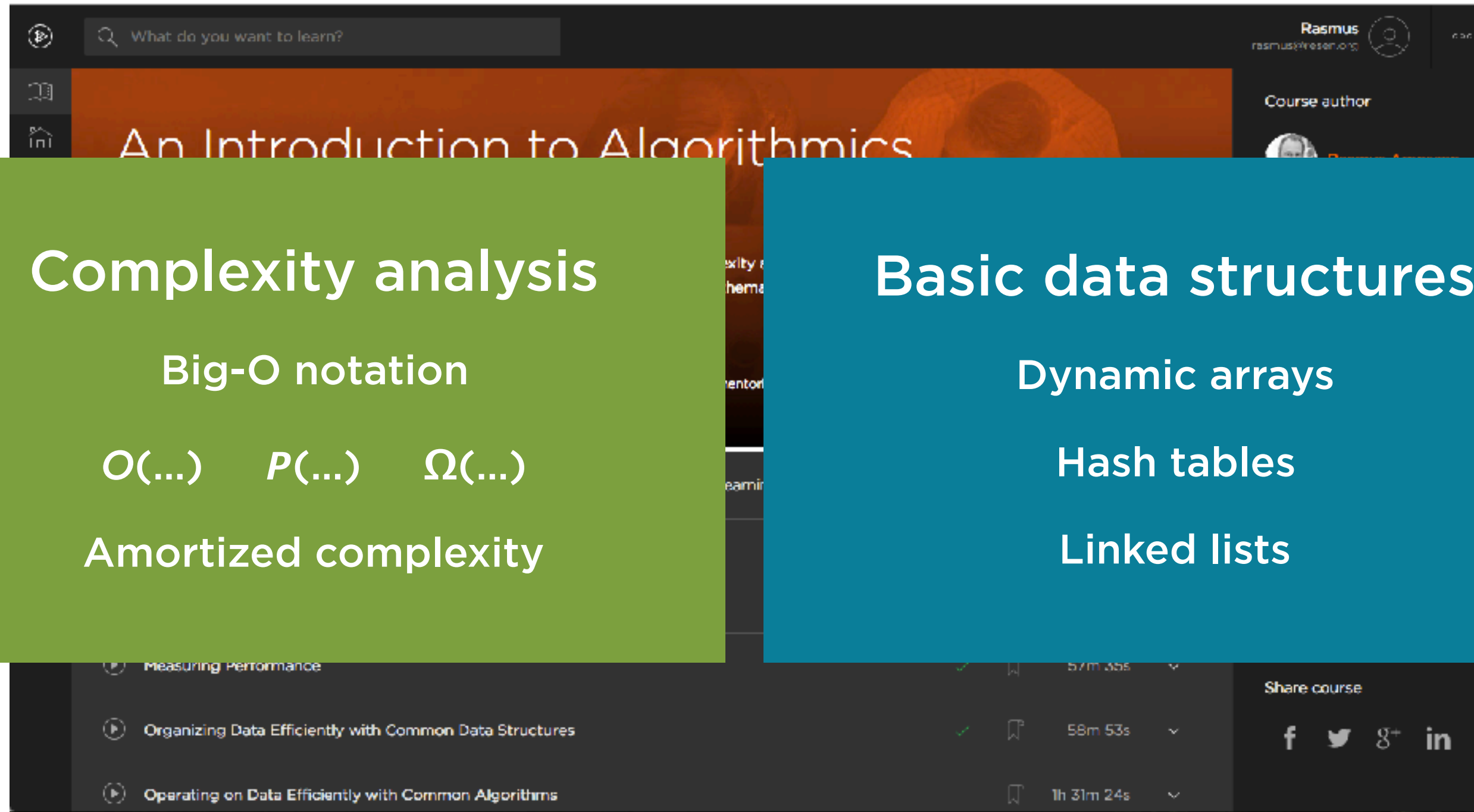
Amortized complexity

Basic data structures

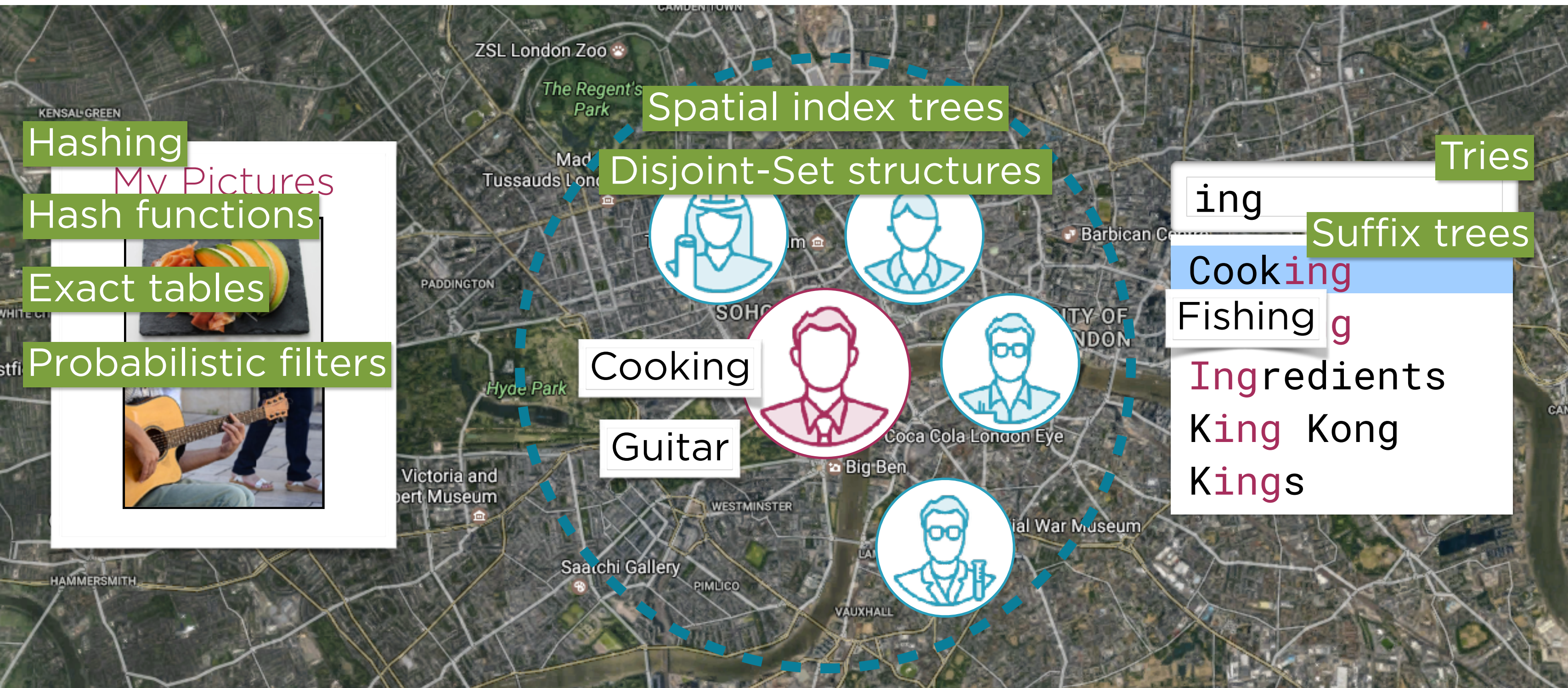
Dynamic arrays

Hash tables

Linked lists



The Match Finder App



How to Watch the Course

Hashing 1: Core Concepts and Exact Matches

Hashing 2: Saving Space with Probabilistic Bloom Filters

Hashing 3: Saving Space and Allowing Deletion with Cuckoo Filters

Efficient Set Operations with Disjoint-Set Structures

Querying Space and Time with Tree Structures

Prefix-querying Sequences Efficiently with Tries

Infix-querying Sequences Efficiently with Suffix Trees