DIT181: Data Structures and Algorithms

Summing Up

Gül Calikli

Email: calikli@chalmers.se

Basic ADTs

- **Queue:** add to one end, remove from the other
- **Stack:** add and remove from the same end
- **Priority queue:** add, remove minimum

Implementation

- A binary search tree
  - Good performance if you can keep it balanced: O(log n)
  - Has good random and sequential access: the best of both worlds
- A hash table
  - Very fast if you choose a good hash function: O(1)

- Assignment3: Question 18 on page 776 on International Edition

Implementing queues, stacks,priority queues

- Queues:
  - a circular array
- Stacks:
  - a dynamic array
- Priority queues:
  - a binary heap

What we have studied

- The data structures and ADTs above
  - algorithms that work on these data
- structures (sorting, Dijkstra's, etc.)
  - complexity
  - data structure design (invariant, etc.)

Data structure design

- First, identify what operations the data
- structure must support
  - Often there's an existing data structure you can use
- Or perhaps you can adapt an existing one?
- Then decide on:
  - A representation (tree, array, etc.)
  - An invariant
- These hopefully drive the rest of the design!

The Written Exam

- The exam
- Friday 16th of March, 08:30 – 12:30,
- Lindholmen

The Exam

- You can bring one A4 piece of paper hand-written only on one side.
- 8 questions
- Total 130 points (100 points + 30 Bonus points)
- Grading:
  - $0 \leq U < 50$
  - $50 \leq G < 75$
  - $75 \leq VG \leq 130$