Algorithm Theory

06 – Amortized Analysis

Dr. Alexander Souza

- Technique for the analysis of algorithms and datastructures
- Works if a sequence of operations has to be analyzed
- How much does an operation cost "on average"?

Amortization

- Consider a sequence $a_1, a_2, \ldots, a_n$ of n operations performed on a data structure D
- T_i = execution time of a_i Could be high.
- $T = T_1 + T_2 + \ldots + T_n$ total execution time
- The execution time of a single operation can vary within a large range, e.g. in $1,\ldots,n$, but the worst case does not occur for all operations of the sequence.
- Average execution time of an operation is small, even though a single operation can have a high execution time.

$$\frac{1}{n} \cdot \sum_{i=1}^{n} T_i$$

Goal: What is the average execution time an operation in a sequence of operations?

Analysis of algorithms

- Best case ⇝ often too optimistic
  smallest execution times
- Worst case ⇝ often too pessimistic
  largest execution times
- Average case ⇝ often realistic distributions unknown, not analyzable often
  Input is drawn from prob. distribution, e.g. all inputs are equally likely
- Amortized worst case

What is the average cost of an operation in a worst case sequence of operations?

Amortization

Idea:

- Pay more for inexpensive operations
- Use the credit to cover the cost of expensive operations

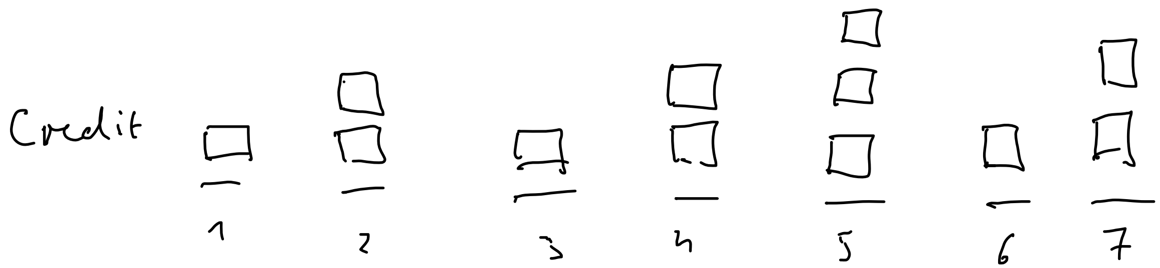
Three methods:

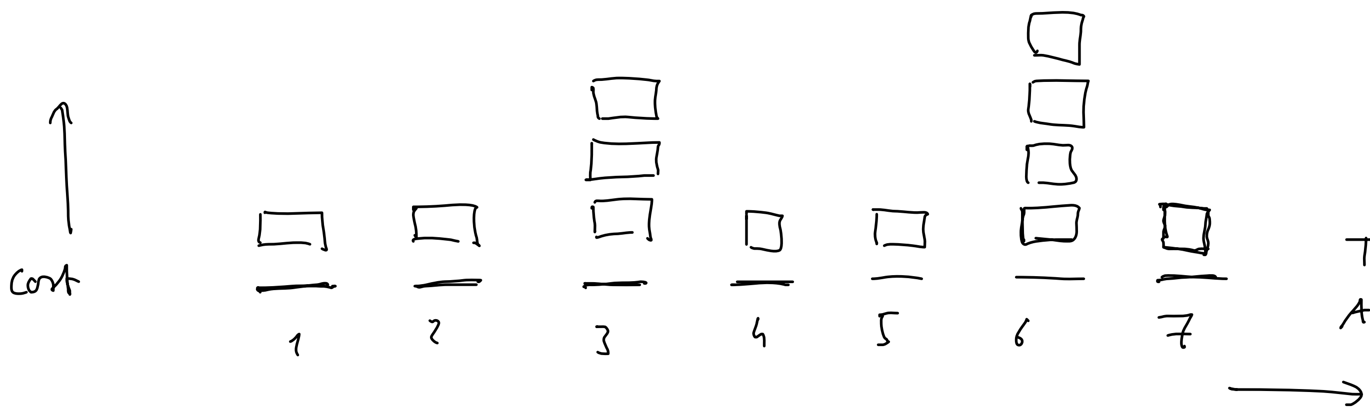
1. Aggregate method
2. Accounting method
3. Potential method

Amortization = Overcharging + Bookkeeping

Pay two ~~cost~~ cost-units per operation

Store unused cost-units in a bank account





Total 12

Average cost $\frac{12}{7} \leq 2$

time