

# How to write a technical report

Magnus Stenhaug

# Technical reports

- A technical report complements and describes the work which has been done
  - Contribution and lessons learned
  - To enable the reader to gain insight and to reproduce the work
- Structured report
  - Follows some basic principles (which is covered in this presentation)

# Introduction

- Gives a brief introduction to the problem
  - E.g. “In this paper, we describe the design and implementation of a set ADT.”
- A subsection can state the requirements for the particular assignment/problem
- The introduction should not give away to many details or discoveries (this comes later)

# Technical Background

- Covers the topics needed to solve the problem
  - E.g. a high-level description of linked-lists and/or trees
  - “What you needed to know before solving the problem”
- Does **NOT** explain your solution
  - This is covered in the design and implementation sections

# Design

- “How did you solve the problem?”
- Give a detailed description of your solution
  - E.g. that set union was implemented using a merge algorithm (along with a description of the algorithm)
- Should **NOT** contain references to actual code
  - I.e. the solution should be language (C, Python etc.) and platform (Windows, Linux) agnostic
- Figures are essential

# Figures

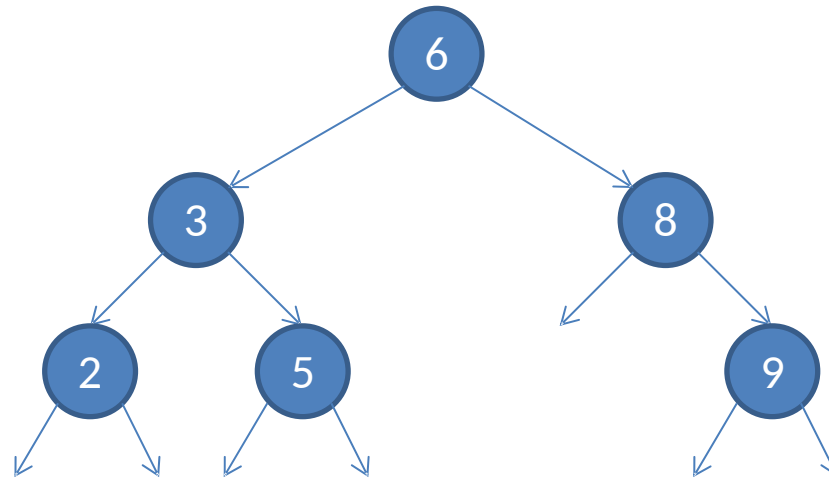


Figure 1: A binary search tree

# Implementation

- Technical details to how the problem was solved (briefly) and (can) contain the answer to:
  - Which language was used?
- This is **NOT** a step-by-step explanation of the implementation
  - As the report is a high-level description of how a problem was solved
  - Don't copy and paste code into the report

# Discussion

- Discuss the design choices made
  - In favor of your solution
  - Compare to other solutions (e.g. trees vs lists)
- Non-functional aspects
  - Performance, scalability, robustness etc.
  - Use Big-Oh notation to describe the algorithms



# Evaluation

- Prove that your implementation is correct and performs as it should
- Correctness checking
  - Validating that the behavior and output is correct given some input
- Performance
  - Metrics such as time, storage cost and memory consumption
  - Compare your solution to others
  - Graphs (e.g. insertion time in list with increasing size)

# Conclusion

- Sum up by restating the problem and solution
  - E.g. “In this report we have described a set ADT.”
- Follow up with a brief summary of the solution along with lessons learned
  - E.g. “By using lists, we were able to implement set operation in linear time”

# References

- Include references to books and other literature used
- DO NOT copy parts of another text into your report

# Language

- Be serious and objective
  - “Since I haven’t learned about mergesort yet ... “
- Avoid adjectives (implies a subjective opinion)
  - “My solution to set union is **extremely** fast”
  - Hard to quantify
- Be formal and precise

# Hints and tips

- Structure is key
  - Each section covers different things
- Headlines first
  - Outline the different sections and subsections and their content
- Describe the solution so that a reader will understand and be able to reproduce it
  - Think of it as a description of your development process

# Tools

- Two choices:
  - LaTeX
  - Editor (Microsoft Word, Google Docs etc.)
- A Word and LaTeX template is provided for you
  - Highly recommended to use and follow these

# Summary

- Structure (**red** sections are most important)
  - Introduction (define the problem)
    - Requirements
    - Technical background
  - **Design (describe the solution)**
  - Implementation (how was it specifically solved)
  - **Discussion (why did you choose this solution?)**
    - Evaluation
  - Conclusion