DON'T

PANIC!

The Student's Guide

through the Exam

- The lecture was about understanding distributed computing models and designing provably efficient algorithms for fundamental problems. Guess what the exam will be about!
- Being able to reproduce the machine code of every algorithm from the lecture is nice, but optional. You will benefit most from the basic concepts and techniques of the algorithms, and knowing the reasons why they work.
- Know the important definitions and their consequences by heart. They shape your distributed world! There are also the ones which don't, they are not important. I know you can tell the difference!
- Praise the **Algorithmic Thinking**! It will protect you in times of exams. The best way to worship it is by solving the exercises (preferably free-style); practicing the rites written in the exam from last year is also holy. Don't be scared if you make flaws, even non-perfect prayers get points, in particular if they reason well about your sins.
- Bringing lots of stuff is also nice, but optional. Of course books are very wise, but you won't have the time to read them front to back! Maybe a "cheat" sheet subsuming the information you consider vital is useful, but it still replaces neither your wits and cunning nor your memory!
- In the exam, do neither haste nor get caught by single questions. It is unlikely that you will have sufficient time for the perfect solution (if such a thing exists[1]). In the beginning, get an overview of the questions and ask your enlightened assistants if something is unclear. Then answer the questions which you can address quickly first. Don't hesitate to skip a part of an exercise if you can do others! Most of them can be solved independently (honestly, believe me!). Do not forget to reserve some time to check your answers; correcting oversights may save you more points than you get by not solving the remaining difficult tasks in a rush.
- Bring a towel. Just in case.

[1] It is widely believed that its mere existence would immediately alter the exam into a more difficult and complex one.