HPPS Epilogue

Troels Henriksen (athas@sigkill.dk)

DIKU
University of Copenhagen

Course Evaluation

Exam Hints

This Year's Exam

The numbers

Scheduling

- Vi havde en hel dag med øvelsestimer så man rigtig kunne sætte sig in i det uden afbrydelser.
- Ønsker at øvelsestimerne var spredt over flere dage så det er nemmere at sprede ens arbejde (øvelsesopgaver + aflevering) over ugen.
- bad that both exercise sessions are on the same day and late in the week. would be better to have one on monday or tuesday and one on thursday, so if we have questions about the assignment we can get help early in the week instad of having to wait until thursday. that way we won't have to use the weekend to complete the assignments…

Feedback

- Feedback har været mystisk i og med, at vi har kunne lave alt fagligt i en opgave, men samtidig har formået at miste 50% point på eksempelvis at inkludere en .git fil, sammensætte to spørgsmål eller tilføje yderligere test-framework. Dertil har det ikke været muligt at rettet igennem for denne feedback, da feedback-processen på afleveringerne har været langsom.
- Lidt bedre feedback på afleveringer ville være nice
- Beskrivelser af opgaver og struktur på afleverings opgaver/rapport er lidt uklar, og der er blevet trukket point for det.

Assignments

- Mange af opgaverne handlede om at forstå kode givet også skrive kun en lille smule selv.
- De obligatoriske opgaver indeholdte i stigende grad kode i handouts som man skulle kunne læse og forstå.
- Øvelsesopgaver var en god måde at forberede sig til opgaverne fordi man havde nogle kode eksempler at kigge på i tilfælde af at man sidder fast.
- **Drop at have både øvelser og afleveringer, man kan ikke nå at løse begge dele, så enten bliver det lidt halvt af hver, alternativt når man ikke at løse øvelserne. Så hellere lav afleveringerne lidt længere.**
- Jeg kan rigtig godt lide at øvelserne der bliver udbudt er meget relateret til afleveringsopgaverne, og at forelæsningerne i høj grad forbereder én til øvelserne.

GitHub

+ Det fungerede utroligt godt med vores GitHub. Det var nemt at finde information.
+ Synes github var nice
- Drop at brug github som kursusrum, det er alt for uoverskueligt i forhold til absalon
+ Github-repo fungerede fint.
- its really confusing that everything on GitHub instead of absalon. Why?
+ Fine setup with the GitHub course page
- Jeg synes at brugen af en GitHub som Absalon side fungere ok, men tilføjer endnu et lag abstraktion i forhold til at finde det relevante information.
- Drop at brug github som kursusrum
+ Jeg kan rigtig godt lide brugen af et GitHub-repository

Anything I missed?

Behind the scenes

https://sigkill.dk/writings/teaching.html

Course Evaluation

Exam Hints

This Year's Exam

What is the purpose of an exam?

- A way for you to demonstrate your achievement of course learning goals.
- `https://kurser.ku.dk/course/ndab20001u/2024-2025`

You are graded on your demonstration of these learning goals.

How this happens in practice

The HPPS exam is a take-home programming problem where you have to demonstrate your *skills* through programming, and your *knowledge* by analysing your code and reflecting upon its properties, by answering specific questions.

- Form is quite similar to assignments.
- A mixture of tasks.
  - Something about reading/writing file formats.
  - Something about a basic sequential algorithm.
  - Something about parallelising it.
  - And perhaps surprises.
  - Some tasks are tightly constrained and come with existing skeleton code (less than in the assignments).
  - Other tasks are more open and will require you to design (small) programs from scratch.

Implications for questions

- These are not trick questions, although some may be subtle.
- *Always* asked in the context of your code—these are *not* general questions about course material.
  - But not *just* about your runtime results—some questions may be hypothetical ("why *might X* be (in)efficient?").
- Some questions are guaranteed to be present:
  - "How did you parallelise *X*?"
  - "What is the temporal and spatial locality of *Y*?"
  - "Show speedup and report scalability of *Z*."
- Answer questions precisely and succinctly.
  - Nonsense raises doubts about your mastery of the learning goals, *even if you also say the right thing along the way*.
  - You don't need to restate parts of the course curriculum, you just need to use it.
  - **Let us look at a reference solution.**

The code is also important

Forms the basis for your questions.
You are evaluated both on correctness, performance, and style.

Regarding style

- Use consistent indentation.
- Don't leave in blocks of commented-out code.
- Don't leave in noisy debug prints.
- Don't leave in unnecessary files.
- Cleanly deallocate memory.

Style is not the most important part, but it *will* have an impact on your assessment.

Course Evaluation

Exam Hints

This Year's Exam

Following information is non-normative.

Counting histograms

Intuition

Given some collection of numbers, compute how many times each number occurs.

$$H[j] = \#\{i \mid x_i \in X \wedge x_i = j\}$$

- $H[j]$ is the number of occurences of j.
- H has k elements and can be represented as size-k array.
- Out-of-bounds elements ignored.
- H can be computed sequentially and with a little more work in parallel.
- Computation cost can depend on both n and k.

Pseudocode for counting histograms

Direct translation of formula:

```
for j < k:
  count = 0
  for i < n:
    if X[i] == j:
      count = count + 1
  H[j] = count
```

Perhaps more efficient:

```
# Assuming H[j]==0 for all j.
for i < n:
  j = X[i]
  if j >= 0 && j < k:
    H[j] = H[j] + 1
```

All-pairs distances histogram

Intuition

Given some collection of points, compute histogram of all distances (rounded down to integers) between points.

A k-bin all-pairs distances histogram of a set of n points P is a k-element array where

$$H[j] = \#\{(a,b) \mid p_a, p_b \in P \wedge a \neq b \wedge j = \lfloor d(p_a, p_b) \rfloor\}$$

where $d(a,b)$ is the Euclidean distance function given by

$$d(a,b) = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2}$$

- $H[j]$ is the number of occurences of distance j.
- Just a counting histogram of all $n(n-1)$ distances.

Histograms for non-numeric data

- Downside of representing *H* as array is that “keys” (the things we count) must be integers.
- A **mapping** associates strings with numeric keys, and can be used to convert collections of strings into collections of integers and back again.
- Counting histograms can then be done on the keys, and the results then converted back into strings.

You will be implementing such a data structure.

Questions?

Otherwise, good luck and see you next year.