Data structures and algorithms, exam

(DAT038, DIT182, DAT495, DAT525, and TDA417)

Thursday 2025-01-16, 14:00–18:00

Teachers:	**Peter Ljunglöf, 0766-075561** **Hazem Torfah, Christian Sattler, Jonas Duregård**
	We will visit exam rooms continuously throughout the exam.

Notes: *Answer directly on the question sheets!*

If you need additional space, you can use extra pages.
Refer to them from the question sheet so that we don't miss them.

Write your anonymous code (not your name) on the first page of the question sheet (and on all extra pages). Don't tear pages out of this question sheet.

You may answer in English or Swedish.

Excessively complicated answers may be rejected.
Write legibly, we need to be able to read your answer!

Questions and solutions will be published afterwards here:
https://github.com/ChalmersGU-data-structure-courses/past-exams

Allowed aids One *hand-written* A4 sheet of notes (double sided).
If you make use of a sheet, you must hand it in along with your answers.

Review When the exams have been graded, they will be available for review in the CSE Student Office at Johanneberg (EDIT floor 6) during their opening hours. If your course was originally given at Lindholmen (DIT18X, DAT495), the exams will be moved after a couple of weeks to the Lindholmen Student Office in Jupiter floor 4.

If you want to discuss the grading, you can contact the examiner via email. Remarks about the grading should be explained thoroughly. Attach a picture of your answer.

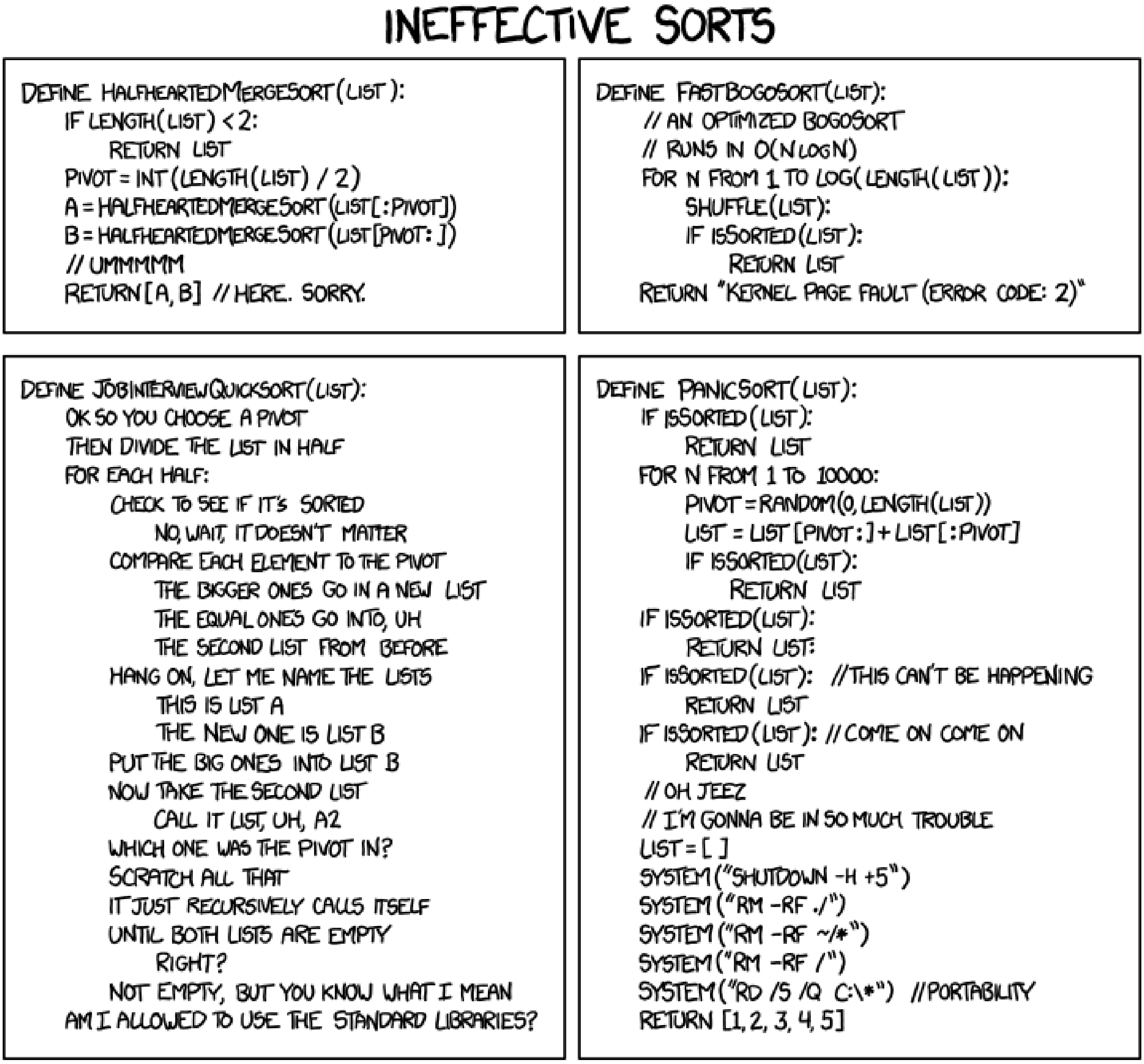
Grading The six basic questions are graded either **correct** or **incorrect**,
while the three advanced questions are awarded 0, 1 or 2 points each.

- To pass the exam, you must pass 5 of the 6 basic questions.

For advanced grades, the following additional requirements must be met:

- 4: 2 out of 6 points for the advanced questions.
- 5: 4 out of 6 points for the advanced questions.
- VG: 3 out of 6 points for the advanced questions.

Good luck!



```
DEFINE HALFHEARTEDMERGESORT(LIST):
    IF LENGTH(LIST) < 2:
        RETURN LIST
    PIVOT = INT(LENGTH(LIST) / 2)
    A = HALFHEARTEDMERGESORT(LIST[:PIVOT])
    B = HALFHEARTEDMERGESORT(LIST[PIVOT:])
    // UMMMMM
    RETURN [A, B] // HERE. SORRY.
```

```
DEFINE FASTBOGOSORT(LIST):
    // AN OPTIMIZED BOGOSORT
    // RUNS IN O(N LOGN)
    FOR N FROM 1 TO LOG(LENGTH(LIST)):
        SHUFFLE(LIST):
        IF ISSORTED(LIST):
            RETURN LIST
    RETURN "KERNEL PAGE FAULT (ERROR CODE: 2)"
```

```
DEFINE JOBINTERVIEWQUICKSORT(LIST):
    OK SO YOU CHOOSE A PIVOT
    THEN DIVIDE THE LIST IN HALF
    FOR EACH HALF:
        CHECK TO SEE IF IT'S SORTED
            NO, WAIT, IT DOESN'T MATTER
        COMPARE EACH ELEMENT TO THE PIVOT
            THE BIGGER ONES GO IN A NEW LIST
            THE EQUAL ONES GO INTO, UH
            THE SECOND LIST FROM BEFORE
        HANG ON, LET ME NAME THE LISTS
            THIS IS LIST A
            THE NEW ONE IS LIST B
        PUT THE BIG ONES INTO LIST B
        NOW TAKE THE SECOND LIST
            CALL IT LIST, UH, A2
        WHICH ONE WAS THE PIVOT IN?
        SCRATCH ALL THAT
        IT JUST RECURSIVELY CALLS ITSELF
        UNTIL BOTH LISTS ARE EMPTY
            RIGHT?
        NOT EMPTY, BUT YOU KNOW WHAT I MEAN
    AM I ALLOWED TO USE THE STANDARD LIBRARIES?
```

```
DEFINE PANICSORT(LIST):
    IF ISSORTED(LIST):
        RETURN LIST
    FOR N FROM 1 TO 10000:
        PIVOT = RANDOM(0, LENGTH(LIST))
        LIST = LIST[PIVOT:] + LIST[:PIVOT]
        IF ISSORTED(LIST):
            RETURN LIST
    IF ISSORTED(LIST):
        RETURN LIST:
    IF ISSORTED(LIST):  //THIS CAN'T BE HAPPENING
        RETURN LIST
    IF ISSORTED(LIST): //COME ON COME ON
        RETURN LIST
    // OH JEEZ
    // I'M GONNA BE IN SO MUCH TROUBLE
    LIST = [ ]
    SYSTEM("SHUTDOWN -H +5")
    SYSTEM("RM -RF ./")
    SYSTEM("RM -RF ~/*")
    SYSTEM("RM -RF /")
    SYSTEM("RD /S /Q C:\*")  //PORTABILITY
    RETURN [1, 2, 3, 4, 5]
```

https://www.xkcd.com/1185/