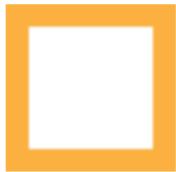


# Algorihms & Data Structures

Text Mining Tries, KD-Trees & Workshop Week 1

Jacob Trier Frederiksen & Anders Kalhauge



cphbusiness

Spring 2019

# Program, Week 18

1. Test Exam & solution suggestion; hand-back
2. Overview: the last weeks of the course
3. Final Exam: information and clearing occasional questions
4. Project 5:
  1. overview
  2. choosing project
  3. hand-out
5. Two last theoretical bits:
  1. Text Mining, 'Tries' (Ch. 5.2)
  2. kD-Trees, (*con amore* -- but part of one project 5 choice)

# Program, Week 18

1. Test Exam & solution suggestion; hand-back
2. Overview: the last weeks of the course
3. Final Exam: information and clearing occasional questions
4. Project 5:
  1. overview
  2. choosing project
  3. hand-out
5. Two last theoretical bits:
  1. Text Mining, 'Tries' (Ch. 5.2)
  2. kD-Trees, (*con amore* -- but part of one project 5 choice)

# Test Exam & solution suggestion; hand-back

- NB: This was an exercise, which we decided to do
- **NB: It is not graded**
- NB: Meant only as an *indicator* of your standing on knowledge
- **NB: There is still time before the finals**

1.4

```
static double q4(int n) {
    int r = 0;
    for (int i = 0; i < n; i++) {
        for (int j = i; j < n; j++) {
            r++;
        }
    }
    return r;
}
```

$$\mathcal{O}\left(\frac{n^2}{2}\right) \equiv \mathcal{O}(n^2)$$

$$n \cdot \frac{n}{2} \approx \frac{n^2}{2} \approx n^2$$

## 2 Improving $\mathcal{O}$ of a slow algorithm

The "ThreeSum" algorithm from the "Booksites" (i.e. 'Algorithms, 4th Edition', Sedgewick/Wayne, Princeton), has a particular scaling in terms of  $\mathcal{O}$ , for  $n$  (compare with the code snippet in 1.4 above for q4). – What is that scaling?

Scaling is  $\mathcal{O}(n^3)$ ,  
There is a faster version of ThreeSum, which can obtain a significantly better scaling (in the variable  $n$ , again). "ThreeSumFast"

- What is the name, and scaling of the faster version of ThreeSum? Scaling:  $N^2 \log N$
- What are the modifications to the slow ThreeSum that makes this improvement possible? Add sort, add binary search  
 $\Rightarrow$  only ~~two nested loops~~ one nested loop.

## 3 Experimental scaling, $\mathcal{O}$ time complexity

For a computing time scaling experiment, an algorithm (unknown to us), exhibited a scaling with respect to  $n$  for an algorithm with the running time as stated below.

$n$	time
125	0,03s
1.000	1,00s
8.000	32,00s
64.000	1.024,00s
512.000	32.768,00s

$\Rightarrow$  Do what is stated in the hint.

Hint: remember our Excel plotting exercise from a few weeks ago. You can find help in the course reno (Week1\excel file), or in the book (ch. 1.1). Plot the numbers

# Program, Week 18

1. Test Exam & solution suggestion; hand-back
2. Overview: the last weeks of the course
3. Final Exam: information and clearing occasional questions
4. Project 5:
  1. overview
  2. choosing project
  3. hand-out
5. Two last theoretical bits:
  1. Text Mining, 'Tries' (Ch. 5.2)
  2. kD-Trees, (*con amore* -- but part of one project 5 choice)

# Overview: the last weeks of the course

– ‘Looking for the summer...’ [Chris Rea]

Date	Week	Teacher	Study Points	Reading, NB: due date!	Main subject	Subject and Goal
16-4-2019	16	-	-	-	-	EASTER HOLIDAY
23-4-2019	17	JTF	-	-	Mid term. + Exam.	Midterm: Course Evaluation and Examination.
30-4-2019	18	AKA/JTF	5.2		Con Amore	Tries / KD-Trees, Project 5 selection and definition.
7-5-2019	19	JTF/AKA	20	5.5	Con Amore	Work on Project 5, and student selected topics / or we choose.
14-5-2019	20	JTF (AKA)			Con Amore, written exam.	Work on Project 5, and student selected topics / or we choose.
21-5-2019	21	JTF	10		Group presentations on a given topic.	Topic within entire syllabus, training presentation skills
<b>03/06/2019</b>		<b>Final exam</b>				

# Program, Week 18

1. Test Exam & solution suggestion; hand-back
2. Overview: the last weeks of the course
3. Final Exam: information and clearing occasional questions
4. Project 5:
  1. overview
  2. choosing project
  3. hand-out
5. Two last theoretical bits:
  1. Text Mining, 'Tries' (Ch. 5.2)
  2. kD-Trees, (*con amore* -- but part of one project 5 choice)

# Final Exam: information

## Information from the Course Catalogue

<b>Algorithms and Data Structures (elective course)</b>
<b>Timing:</b> Spring - 1 <sup>st</sup> /2 <sup>nd</sup> semester
<b>Scope:</b> 10 ECTS
<b>Contents:</b> The purpose of the module is to make the students able to explain and implement simple and complex data structures and to use algorithms to manipulate these. Further, equip students for designing their own algorithms to solve a given problem. Use of time and memory and scalability will be analysed.
<b>Learning Objectives:</b>
<i>Knowledge</i>
The graduate will possess knowledge of:
<ul style="list-style-type: none"> <li>basic data structures such as stacks, queues, heaps</li> <li>binary trees, balanced trees, and hash tables</li> <li>basic sorting and searching algorithms</li> <li>basic graph theory</li> </ul>
<i>Skills</i>
The graduate will be able to:
<ul style="list-style-type: none"> <li>select and use a suitable data structure for a given task</li> <li>select and use suitable sorting, searching, and hashing routines</li> <li>select and use algorithms for graphs</li> <li>analyze algorithms for consumption of time, memory and scalability</li> </ul>
<i>Competencies</i>
The graduate will be able to:
<ul style="list-style-type: none"> <li>select, scale test, and use appropriate algorithms for practical problems</li> <li>devise, develop, scale test, and use algorithms for practical problems</li> </ul>
<b>Examination form:</b> Written test (pass/fail), and oral exam based on the student presentation and the syllabus.
<b>Assessment:</b> One single grade is given according to the 7-point grading scale.
<b>Admission criteria:</b> Min. 80% study points obtained. 80% of all mandatory assignments handed in. The written test passed.
<b>Examination basis:</b> all contents of the course syllabus, and mandatory hand-ins.
<b>Consequences of not passing exam:</b> Re-exam (limited number of attempts).

# Final Exam: clearing occasional questions

## Exam details

<b>Algorithms and Data Structures (elective course)</b>	<b>10 ECTS</b>	<b>Internal</b>
<b>Timing:</b> ultimo Spring semester 2019		
<b>Examination form:</b> The exam is oral but with part of the exam as a written (pass/fail) test in the end of the course. For the oral part, the student will prepare a presentation (max. 5 mins) of the solution of one of the major hand-in assignments. Further examination/discussion (max. 15 minutes) will be based on the presentation, but can include all aspects of the curriculum.		
<b>Assessment:</b> One single grade is given according to the <u>danish</u> 7-point grading scale.		
<b>Admission criteria:</b> Min. 80% study points obtained. 80% of all mandatory assignments handed in. Written test passed.		
<b>Consequences of not passing the exam:</b> Re-exam (limited of number of attempts).		

## Study Points

### Algorithms and data structures

Dates for exam attempt and OLA:

- 1<sup>st</sup> attempt: xx. 2019 9.00 AM
- 2<sup>nd</sup> attempt: yy. 2019 9.00 AM
- 3<sup>rd</sup> attempt: zz. 2019 9.00 AM

Mandatory learning activity	Description	Study Points
Hand-in assignments Written test Give a lesson	Of which >= 80% must be completed Pass/fail test; must pass for oral exam eligibility On a given topic in small groups	Total of 100 (equally weighted) - 10

# Program, Week 18

1. Test Exam & solution suggestion; hand-back
2. Overview: the last weeks of the course
3. Final Exam: information and clearing occasional questions
4. Project 5:
  1. overview
  2. choosing project
  3. hand-out
5. Two last theoretical bits:
  1. Text Mining, 'Tries' (Ch. 5.2)
  2. kD-Trees, (*con amore* -- but part of one project 5 choice)

# Project 5 – reaching a level of independence...

You get to choose your topic (from a list of four):

1. **Out-of-core Sorting** of ‘Big Data’ (by volume, not heterogeneity)
2. **Text Mining**, Tries, sorting of ‘Big Data’ (by variety, and volume)
3. **Error detection** in sorting of ‘Big Data’ (by volume, not heterogeneity)
4. **kD-Tree acceleration**, searching kD spaces w/applications

# Program, Week 18

1. Test Exam & solution suggestion; hand-back
2. Overview: the last weeks of the course
3. Final Exam: information and clearing occasional questions
4. Project 5:
  1. overview
  2. choosing project
  3. hand-out
5. Two last theoretical bits:
  1. Text Mining, 'Tries' (Ch. 5.2)
  2. kD-Trees, (*con amore* -- but part of one project 5 choice)

# Text Mining

- ‘Tries’ by Anders Kalhauge
  - Relevant in particular for project 5 on text mining
  - Relevant generally for exam and in real ”IT Life” ☺

# Program, Week 18

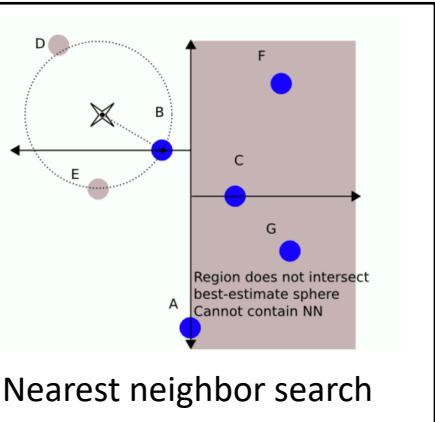
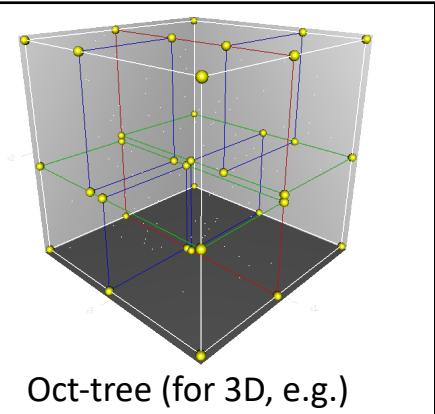
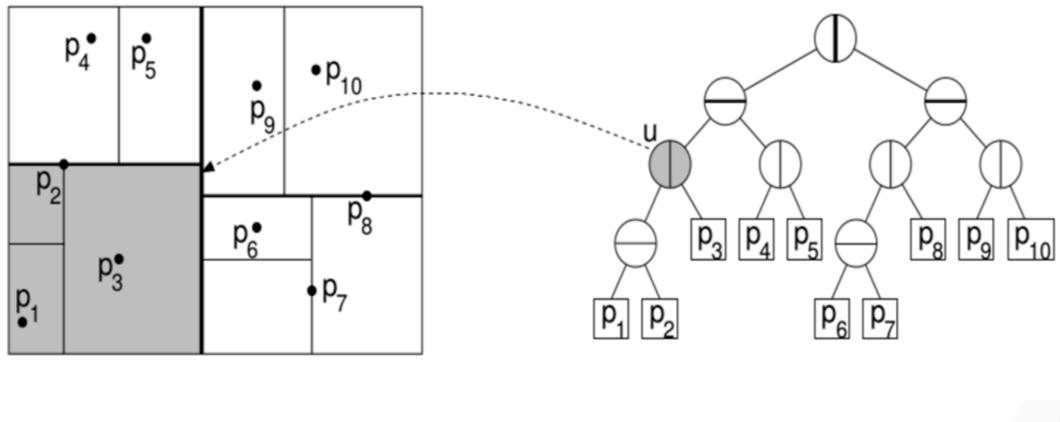
1. Test Exam & solution suggestion; hand-back
2. Overview: the last weeks of the course
3. Final Exam: information and clearing occasional questions
4. Project 5:
  1. overview
  2. choosing project
  3. hand-out
5. Two last theoretical bits:
  1. Text Mining, 'Tries' (Ch. 5.2)
  2. kD-Trees, (*con amore* -- but part of one project 5 choice)

# kD-Trees (here just 2D but more general)

(*con amore*, but part of project 5 option)

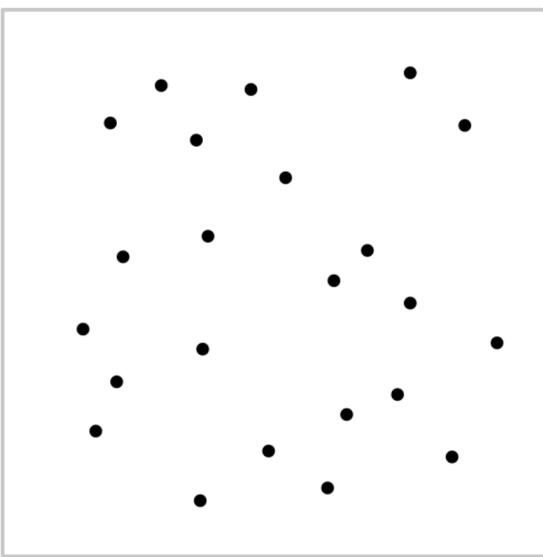
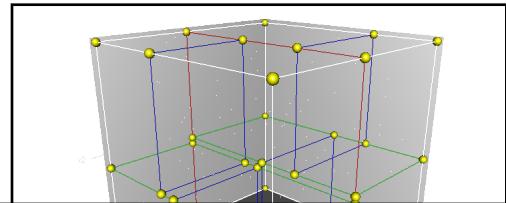
- This is the first of two formal slides on kD-Trees, today.
- The rest will be free wheeling, or “*winging it*”, with some interesting applications in sight.

Quad-tree (2D, e.g.)

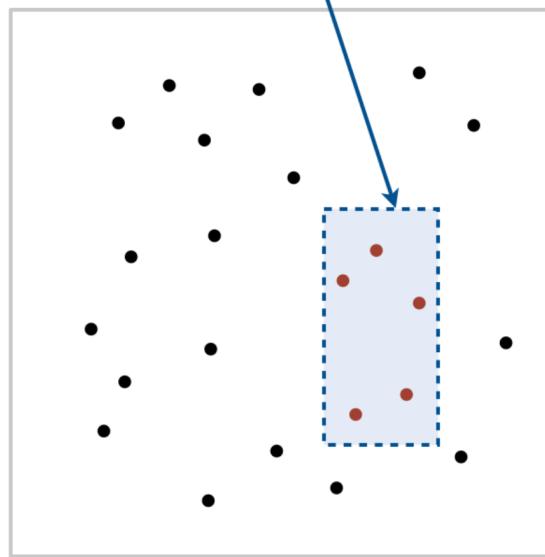


# kD-Trees (here just 2D but more general)

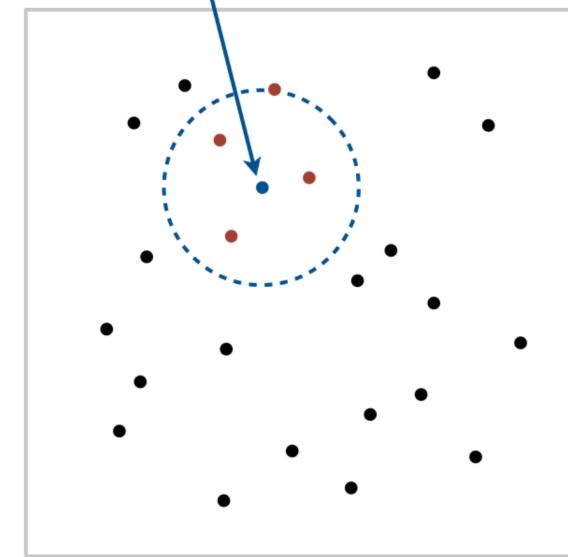
(*con amore*, but part of project 5 option)



**set of points**



**2d range search**



**4 nearest neighbors**