Grading Criteria for
DD2440 Advanced Algorithms
2018
Assessment tasks

• **3+1 individual assignments.**
  – **3 normal assignments.** Each assignment consists of 5-6 questions. Each question is assigned a grade between A-E.
  – **1 make-up assignment,** consisting of 1-2 questions only of grade E.
  – To be completed individually.
  – Grade for each question: pass/fail

• **3 group exercises.**
  – To be completed in a group of 2-4.
  – Possible to consult the internet, but cannot copy a solution from somewhere.
  – Grade for each question: pass/fail

• **2-3 Projects.**
  – To be completed in a group of 2-4.
  – Grade: Pass/fail, but possible to get grades A-E for some projects.

• **Peer-reviews:**
  – Review peers’ turn-ins for the above tasks.
  – Required to pass any task above.
  – In this an *individual responsibility*. You have to complete the peer review assigned to you (not your group). Consulting others is allowed though.
Checklist for passing with E

- **Individual assignments**: Pass 2 assignments.
  - To pass an assignment, pass one question from such assignment.
- **Group exercises**: Pass 1 question in total.
- **Projects**: Pass 1 Project
- **Peer-reviews**: To pass any task above, passing the corresponding peer review is mandatory.

*Example*: If you pass individual assignments #1 and #2, group exercise #1, and project #1, then you pass.
Checklist for grade $X \in \{A, \ldots, E\}$

• **Pass the course:** see the checklist for passing the course

• Get grade **at least $X$** from 3 among individual assignment questions* and projects.
  – To get grade $X$ from a individual assignment, pass one question assigned with grade $X$.
  – Some (but not all) projects offer an opportunity to earn grades. Each of them has different rules.

• **Example 1:** If you get A from individual assignments #1 and #2 and project TSP, and pass group exercise #1, then you get A.

• **Example 2:** If you pass A- and B- questions individual assignments #1 and get B for project TSP, and pass group exercise #1, then you get B.

* Possible that questions are from the same assignment
**Bonus**

**Usage:** You can use 4 points to "upgrade" your grade (e.g. from B to A) of one of your individual assignments or projects.

**Remark:** To upgrade your grade for any task, you must receive at least E for such task.

**Earning:** The bonus points could be earned by contributing to the learning of participants in the course. Examples:

- Scribe a note for a lecture (see examples from the previous years [here](#)).
- Find or construct materials or activities (a note, a video, etc.) that help understanding lectures better.
- Provide model solutions that are different from existing ones for group exercises or individual assignments.
- Provide extra practice questions and solutions.

**Remarks:** Please feel free to propose other opportunities yourself. It is possible to earn in group, but must justify the work of 2 hours each. Talk to the lecturer if you want to propose anything.
Intended Learning Outcomes

1. Explain different algorithmic concepts such as randomized algorithms, approximation algorithms, fixed-parameter algorithms
   – Assessment: Individual and group assignments
2. Analyze and verify algorithms that are based on the above concepts
   – Assessment: Individual and group assignments
3. Develop algorithms using the above concepts
   – Assessment: Individual and group assignments, and projects
4. Independently explore existing advanced algorithms, implement them, and improve them using heuristics.
   – Assessment: Projects
5. Communicate algorithmic ideas in a clear, formal, way.
   – Assessment: Individual and group assignments, and project reports

* The term “advanced algorithms” is a bit subjective and students should consult TAs or the lecturer if in doubt.
<table>
<thead>
<tr>
<th>Learning Outcomes</th>
<th>E</th>
<th>D</th>
<th>C</th>
<th>B</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Explain different algorithmic concepts such as randomized algorithms, approximation algorithms, and fixed-parameter algorithms</td>
<td>Explain definitions and principles of different concepts and models of computation</td>
<td>Pick concept or model that is suitable for each situation</td>
<td>[C criterion] with algorithms similar to the simple ones covered in class</td>
<td>[A criterion] with given hints or with less complex problems</td>
<td>Analyze and verify algorithms that are different from those covered in class based on principles covered in class</td>
</tr>
<tr>
<td></td>
<td>Analyze and verify algorithms similar to the simple ones covered in class</td>
<td>[C criterion] with given hints or with less complex problems</td>
<td>Develop algorithms based on the same idea as algorithms covered in class</td>
<td>[A criterion] with given hints or with less novel ideas</td>
<td>Develop algorithms to solve problems using principles covered in class that require some creativity (i.e. ideas that are not covered in class)</td>
</tr>
<tr>
<td>2. Analyze and verify algorithms that are based on the above concepts</td>
<td>Verify and comment on the algorithms and reports written by peers.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Develop algorithms using above concepts</td>
<td>Develop algorithms along the same line of simple algorithms covered in class with proofs</td>
<td>[C criterion] with given hints or with less complex ideas</td>
<td>Develop algorithms based on the same idea as algorithms covered in class</td>
<td>[A criterion] with given hints or with less novel ideas</td>
<td></td>
</tr>
<tr>
<td>4. Independently explore existing advanced algorithms, implement them, and improve them using heuristics</td>
<td>In group and without much external help, survey, evaluate, and implement algorithms to solve assigned problems with sufficient efficiencies.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. Communicate algorithmic ideas in a clear, formal, way.</td>
<td>Clearly and formally explain algorithms, their analysis, and relevant literatures in the solutions and reports.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>