

## Meeting and communication norms

- How often will the team plan to meet outside of class? How long do you anticipate meetings will be? What will you do if things change?

We will try to meet at times that align with office hours so that we have time to debug. We don't have any expectations about the amount of time we will meet, but it will just be until we are satisfied with our performance.

- Where and when will outside-class meetings be held? What will you do if someone fails to show up for a meeting?

We don't have a strict requirement on where, some places might include the stud, Hayden, etc. We will try to meet sometime roughly before office hours so we have time to work on something, go to OH to get ideas / debug, and then implement those ideas. If someone fails to show up, prioritize communication (ideally beforehand).

- How will you communicate outside of meetings?

## Text

- If someone in the group decides to drop the class, what obligations does that person have to inform the remaining team members and complete promised work?

No one is planning to drop the class.

## Work norms

- How much time per week do you anticipate it will take to make the project successful?

We are expecting to do around 13-14 hours a week, but generally we care about our performance rather than the time we spend.

- How will work be divided among team members? On which parts of the project will you do pair programming?

We will start with pair programming mainly, and then eventually we might individually try out optimizations if we have ideas.

- What will happen if someone does not follow through on a commitment, e.g., not doing their work? What if someone gets sick?

We will prioritize communication. In case of larger issues we will communicate with the TAs.

- How will the work be reviewed? How will you manage your code branches?

For pair programming, reviewing will just happen automatically. When we have new ideas, we will make a separate branch for it.

- How will you deal with different work habits of individual team members? (e.g., some people like to get work done early, while others like to work under the pressure of a deadline.)

We will try to get a lot of progress initially and do most of our big optimizations quickly.

### Decision making

- Do you need unanimous consent to make a decision? What process for decision-making will you use if you can't agree?

We will make decisions based on the performance and correctness of the code, so that will ultimately be the main basis of decision making.

- How will you prioritize the work to be done? How will you deal with the common situation in which different team members have different optimization ideas?

Try to start with the simplest optimizations / the recommended ones, and then if have any other ideas we can try them after.

- What happens if everyone does not agree on the level of commitment? (e.g., some team members want an A, but others are willing to settle for a B.)

We will try to aim for an A / being in the highest x performing submissions.

- Is it acceptable for some team members to do more or less work than others?

We understand that at some times one of us might be busy / get sick, so we will try to keep things balanced overall. For example, if one person is busy / sick for the beta submission, they will try to contribute more for the final submission.