# The algorithm

initialize R to be the set of all availabilities;

sort R by increasing finish time;

initialize ordered list A to be empty;

while (R is not yet empty):

get i∈R; #notice i ends the earliest out of all of R

remove i from R;

if ∃ j∈R such that i and j overlap by some time:

get the first j∈R; #notice j ends the earliest out of all of R

add i and j to the end of A;

endwhile

return A as set of all partners (every two consecutive elements are partners)

# Proof of Correctness

We prove this greedy algorithm’s correctness using induction.

## Base Case

Post [218](https://piazza.com/class/jzqwd6s59yh6bm?cid=218) on Piazza clarified that there’s at least one student. So we need to prove P(1). Since you obviously can’t get a pair if there’s only one student, the most number of pairs you can get is 0, the same as what this algorithm returns because there doesn’t exist another availability j∈R that overlaps with i, so A is empty.

## Inductive Case

Assume P(n) to prove P(n+1).

Define P(n) to be “there’s n students and we’ve got the maximum number of eligible pairs in A by running this algorithm.”

First Observe that the maximum number of eligible pairs can be gotten through the greedy algorithm being run on a segmented version of this problem, where each segment boundary in time is a time where no availability crosses, and then adding up the answer from each segment. This is obviously true because eligibility requires overlap. A visual representation of this problem being segmented (different colors mean different segments) :

Notice that any student whose availability doesn’t overlap with anyone else’s is in his own segment, which gives the answer 0 for his segment. We will form our argument on one segment, since the final answer can be gotten by adding up each segment’s answer. Observe that due to the nature of this algorithm, any of the n students that isn’t in a pair is in one of these situations:

1. His availability doesn’t overlap with anyone else’s (he is in his own segment)
2. His availability does overlap with someone, but it’s at the end of his segment after all eligible pairs are removed.

When an additional student is added to the mix of n students to result in n+1 students, the optimal solution then may or may not result in more matchings than P(n)’s number of matchings. Observe here that the newly added student’s availability ends no earlier than all of the original n students’ because R is sorted by finishing time. To increase the number of matches, observe that at least one of the currently unmatched students gets matched to someone, either a student part of the original n students, or the newly added student. Let’s investigate the possibilities in each of the two cases presented above:

1. **A student who wasn’t compatible with any of the original students** (someone in his own segment) would still not be paired with any of the original students, but will get paired with the new student if their availabilities overlap. This would not mess up the original pairs because when the algorithm is run with the n+1 students, the lone student would choose the newly added student, and the new student is the only one who is compatible with the lone student, and so all other students would proceed to choose the same partners as before. This adds one more pair to the mix, proving P(n+1) in this case.
2. **A lone student does overlap with someone, but his availability is at the end of R.**