**Vision Statement**

**Defining System:** The system will perform a fair division of elective courses for the students, according to their preferences over the courses and according to the maximum capacity of each course.

The problem consists of a set of courses, some of which may have more than one section/time offered. Define the set C={1,..,m} as a set of all course sections offered in the market in which each course section j has qj seats. The course section seats are allocated to a set of students S = {1, .., n}, each of whom can take at most k courses in each semester. By standard assumptions of rationality, each student i has real ordinal preferences (utility) over course sections uij.

**Target Audience:** the target audience is for university departments and their students.

**Reason:** Today in most higher education institutions, the mode of distribution of elective courses is in the "first come, first served” (FCFS) method, which means that registration opens at the same time for a large number of students.

In our opinion, this method causes an unfair division of the courses between the students. Thus, we would like to develop a new method of fair division among the courses.

**Similar systems:** These days, several similar systems exist on the market which most of them based on the following algorithm:

1) "first come, first served” (FCFS) - During the application period students can assign to several courses wherever free seats are available based on their time of arrival after the registration time starts. Changing the course assignment later is only possible after the existing course assignment is canceled by a student.

Flaws:

* This model creates a major burden for the server and sometimes it could lead to collapse.
* The registration process for groups of courses is often ordered sequentially, such that students have to decide in one period whether they register for a particular set of courses so that they don’t have enough time to make the smart decision for them.
* In our opinion, the main disadvantage is that this method not provides a fair division.

2) Boston mechanism – in past this method has been common, over the years it has become less common due to its flaws. The Boston mechanism is an algorithm that has been used for school choice, it works as follows: Let us assume that there are m schools. For round k = 1 to m, it does the following. In each round k, those students who have not been allocated a seat yet are considered to be allocated a seat at their k-th most preferred school. The seats are allocated according to the priorities of the schools as long as capacity is not used up. By the end of round m, each student has been allocated a school seat.

Flaws:

* This mechanism does not exhibit a dominant strategy equilibrium
* This algorithm doesn’t fulfill the stable matching principle.

3) Gale & Shapley - an algorithm for finding a solution to the stable matching problem, this algorithm matches between a student S and a curse C so that (S, C) will be a stable match.

Boston mechanism was replaced by Gale Shapley, in most higher education institutions in the U.S and has been widely using in Germany.

Description of the algorithm:

Suppose there are S students, C courses, and K courses that any student need to choose from I have to choose from.

- S Students submit applications to K courses that they most want out of all the courses that have not yet been rejected.

- Any course that has more students than its capacity, will contain the prioritized students for this course and rejects everything else.

- Repeat the first steps until each student has K courses to which each is matched.

Flaws:

1. The algorithm is not necessarily Pareto efficient.
2. In order to use the algorithm, we have to create a preference relation for the courses over students.

4) EADAM:

introduced a matching mechanism, which reduces welfare losses on the student side as described above, but gives up on strategy-proofs.

Description of the algorithm:

Works the same way as Gail Shepley works except for the fact that after one step a pair (s, c) is found so that S represents a student and C represents a particular course. This pair will be called an "interrupting pair" in other words a pair that is certainly not a stable match.

So that if after one step we can see that course C matched to the student S but we know for sure that this combination is not a stable match, then instead of waiting for more students to sign up for this course, we will do the following:We will remove course C from student S's preferences table so that he will not be able to pre-register for this course. it will prevent students that are not "suitable" for this course from signing up for it.

Flaws:

1. Does not provides the strategy-proof principle

ii) In order to use the algorithm, we have to create a preference relation for the courses over students.

5) A-CEEI:   
A-CEEI mechanism was introduced by Budish (2011).

Description of the algorithm:

In A-CEEI, students report their ordinal preferences over all possible schedules of courses, and they have the option to report either their additive or nonadditive complementary/ substitutability preferences, depending on the specific implementation. Then, it assigns a random, approximately equal budget to students and allocates courses through a series of optimizations based on student preferences, student budgets, and course prices, emulating a Walrasian-style price equilibrium. Although several algorithmic variations are possible, one concrete implementation uses a metaheuristic tabu search method to find a nearly optimal price set, which approximately equates the number of available seats of each course (supply) with its demand. The course demand for each price set is calculated by allocating each course seat to a student who wants it and can afford it. The final allocation of courses is achieved when the error, the difference between allocated and available seats, is less than a predetermined tolerance.

Flaws:

1. A-CEEI allows infeasible solutions in which the number of assigned students may exceed the course capacity, a problem to be sorted out after the mechanism.
2. This method randomly prioritizes students with randomly generated budget inequality directly affecting the results, which should, in principle, be avoided whenever possible.
3. In particular, the returned allocation is not Pareto-efficient - some items remain unallocated.

6) DRAFT:  
In the draft mechanism, used by Harvard Business School beginning in the mid-1990s.

Description of the algorithm:

a computer takes students’ preferences over individual courses. Then student proxies take turns in a “draft order” with one available course seat assigned to each student in each round. In the draft-order procedure, students are first randomly ordered, and in even-numbered rounds, their order is reversed (Budish and Cantillon 2012).

Flaws:

1. The draft mechanism only collects the student's ordinal preferences and, therefore, cannot distinguish between a slight difference and a significant difference among preferences. Hence, a student who is almost indifferent between two courses C1 and C2, and has put C1 first may get this course while another student who wants C1 much more than C2 may lose it.
2. Does not provides the strategy-proof principle

7) Custom Hungarian algorithm: (our Idea)

The Hungarianmethod is a [combinatorial optimization](https://en.wikipedia.org/wiki/Combinatorial_optimization) [algorithm](https://en.wikipedia.org/wiki/Algorithm) that solves the [assignment problem](https://en.wikipedia.org/wiki/Assignment_problem) in [polynomial time](https://en.wikipedia.org/wiki/Polynomial_time) and which anticipated later [primal-dual methods](https://en.wikipedia.org/wiki/Duality_(optimization)). It was developed and published in 1955 by [Harold Kuhn](https://en.wikipedia.org/wiki/Harold_Kuhn), who gave the name "Hungarian method" because the algorithm was largely based on the earlier works of two [Hungarian](https://en.wikipedia.org/wiki/Hungary) mathematicians: [Dénes Kőnig](https://en.wikipedia.org/wiki/D%C3%A9nes_K%C5%91nig) and [Jenő Egerváry](https://en.wikipedia.org/wiki/Jen%C5%91_Egerv%C3%A1ry).

Description of the algorithm

1. We multiply the number of students by the number of courses that they need to choose, also we multiply each course by his capacity.
2. And then we will run the Hungarian algorithm.

Flaws:

1. The main problem is clashing between requests with the same preference.
2. There is a possibility of clashing between the selected courses and the courses that we want to sign up for.

8) SP:  
An algorithm-based round-by-round mechanism, however, if a student gets rejected it transform the unspent points to use on the next most preferred course at the same round. This algorithm is probably used as “BIDDING” at Tel Aviv University.

Description of the algorithm:

1. Each student is given a fixed budget (e.g., 1,000 points) and submits bid points bij for each course with the sum not exceeding the budget.
2. Every round, the algorithm acts on behalf of each student as follows

i. Each student updates his preferences table according to the collision of other courses that he already sings up to. Also, removing courses that have a full capacity.

ii. Each student offers a bid amount for his top priority course.  
iii. Each course accepts up to his available seats of the highest offers and rejects any remaining offers. Then capacities are updated according to the new registrants.  
iv) If a student gets rejected by a certain course then he gets return unspent points to use on the next most preferred course.  
Furthermore, the rejected students will repeat steps (i)–(iii) until no more students are rejected.

1. Step 2 repeats k times.

To conclude, this algorithm has a strategy-proof. Also, SP makes a fair division because he is based on a round-by-round mechanism. We haven’t found major flaws in this particular algorithm and thus we will focus on it.

We’ll also try to develop the custom Hungarian method such that it will be a combination of algorithm 7 (custom Hungarian algorithm) and algorithm 8 (SP), but we’ll test it before we got a conclusion on what is the preferable algorithm.

Keywords:

1. Algorithmics of matching under preferences.
2. Course allocation via Stable Matching.
3. A matching model between courses and students.
4. Course allocation problem.

Essential features:

1. Create a system that will provide a fair division of courses among students.

2. Create a system that provides a specific solution for the Department of Computer Science at Ariel University.

3. Basing our project on the main algorithm.

4. The final output of the system will include consideration of colliding courses

Desirable features:

1. The system will provide statistical information on the verge of admission to courses from previous years.

2. The system will be synchronized with the university system, and it will include logging in according to the university standards.

3. Comparing our algorithm with other algorithms according to their output and their runtime.

4. Create a universal system for all departments in the university.

Elevator statements:

Today, the existing system at Ariel University is FCFS which in our opinion is not a fair division.

Our vision is to create a fair division of courses among the students, We would like to optimize the choosing curses process so that all students feel equality and fairness.

The system will provide Early Information such that syllabus, lecturer, date & time, capacity, etc.

Each student will rate the elective courses as he sees it fit, and the system will provide each student with a fair choice for him.

ID: Itay Simhayev – 205666407, Joseph Schtein – 206128613, Lihi Belfer – 204784268.