Discrete Mathematics
Mini-Project

At the Gym

Brief Description

After spending some time at the TCNJ gym, I realized how difficult it really was to workout consistently, without interruption. There are never enough machines for everyone to use when they want and for the amount of time they want to use them for. I started to recognize the discrete math in this situation and used this idea in coming up with an every-day real world problem related to this idea.

Every now and then in the summertime, I like to go to the gym and have a 'Fitness Session' with some of my friends. A few of my friends are friends from the gym, and are always there, and some of them I know from other activities. We all like to use different equipment for different amounts of time, which can often result in a conflict. There are also more of us then there are of machines, so we must manage our time carefully. Besides the equipment available in the gym, there is a small lunch cafeteria and a heated pool that some people can use while taking a break or waiting for equipment. The problem is that some of my friends do not like each other, so these sets of people should not be out socializing at the same time. Therefore, I would like to use discrete mathematics to form a structure for the Gym, and find a sufficient solution as to how we can manage our time in the gym together so that none of us are ever stationary, given that we would like to spend the minimal amount of time there, using our time as efficiently as possible.

I will simplify the structure of a typical gym greatly, using only a few pieces of equipment and only allowing my friends entrance into the gym. This will simplify the topic enough that I will not have huge lists of sets and relations in the problem.

Gym = (*Friends, People, Equipment, Treadmill, Bike, Dumbbells, Cafeteria, Pool, uses, goesOnAfter, fightsWith, timeLength)*

To do this, it was imperative for me to define a number of sets that are involved in this scenario. "Friends" will represent the set of my friends in the gym, "Treadmill" will be the set of available treadmills, "Bike" the set of available bikes, "Dumbbell" the set of available dumbbells, "Pool" the set of pools, and "Cafeteria" the set of cafeterias.

Friends= {Sarah, Claire, Josh, Laural, Tim, Peter, Lauren, Nikki, Noah, Katie, Jo}

Treadmill = { TM1, TM2, TM3}

Bike = { B1, B2}

Dumbbells = {D}

Cafeteria = {cafeteria}

Pools = {smallPool}

Equipment = {Treadmill, Bike, Dumbbells}

Furthermore, since I am not my friend, but am at the gym, there is a set called "People". Logically,

Friends is a subset of People and People-Friends = Me

When my friends and I go to the gym, there are specific pieces of equipment that each of us would like to use. During our short trip to the gym this summer, we would all like to complete our workouts by using our favorite equipment. The relation "uses" in this situation can describe the relation between each person and the equipment that each person would like to use during their visit. It is logical to use a relation in this instance because each person may want to go on more than one piece of equipment.

uses = {(Sarah, Treadmill), (Sarah, Bike), (Claire, Dumbbells), (Josh, Treadmill), (Josh, Dumbbells), (Josh, Bike), (Laural, Bike), (Tim, Dumbbells), (Tim, Treadmill), (Peter, Treadmill), (Peter, Dumbbells), (Lauren, Bike), (Lauren, Dumbbells), (Nikki, Bike), (Noah, Dumbbells), (Katie, Treadmill), (Katie, Bike), (Jo, Dumbbells), (Jo, Bike), (Me, Treadmill), (Me, Dumbbells)}

Type: People X Equipment

In trying to make our trip as short as possible, we need to keep in mind that each activity requires a different amount of time. Therefore, "timelength" is a function describing the amount of time needed on each activity.

timeLength = { (Treadmill, 30), (Bike, 20), (Dumbbells, 10)}

Type: Activities? Minutes

Obviously, taking the different times, desired activities of each friend, as well as the limited number of equipment into account, we can see that not everyone will be able to work out on equipment at the same time. Because of this, the gym has some "common activities" that my friends can engage in when they are not working out: a pool and a cafeteria. However, these are also considered social activities because a number of people group together and share in having lunch or swimming in the pool. Some of my friends do not get along, and in fact fight when they hang out together. Therefore, it is necessary to make sure that those who fight with each other do not engage in these social activities at the same time and place. The relation "fightsWith" describes the friends that fight with each other. As can be inferred, this relation is symmetric because if one person fights with another, the other one will fight back.

fightsWith = { (Tim, Josh), (Josh, Tim), (Laural, Tim), (Tim, Laural), (Nikki, Noah), (Noah, Nikki), (Josh, Jo), (Jo, Josh), (Sarah, Lauren), (Lauren, Sarah), (Claire, Peter), (Peter, Claire)}

Type: Friends X Friends

Solution:

Based on this information, I made a plan outlining a "schedule" of who should be on what equipment at a certain time. In order to make our visit to the gym as short as possible, (as it is very nice outside) I created a function describing how each person would go onto the specified equipment directly after another is finished. This is called "goesOnAfter", and is logically represented as a function because there can only be one person going on a specific machine after another is finished. This person is unique to the time and circumstance.

goesOnAfter = {((Lauren, Me), D), ((Me, Peter), D), ((Peter, Josh), D), ((Josh, Jo), D), ((Jo,

Noah), D), ((Noah, Tim), D), ((Tim, Claire), D), ((Katie, Lauren), B1), ((Lauren, Sarah), B1), ((Sarah, Josh), B1), ((Jo, Laural), B2), ((Laural, Nikki), B2), ((Sarah, Me), TM1), ((Josh, Peter), TM2), ((Tim, Katie), TM3)}

Type : (Friends X Friends) ? Equipment.

With the help of these relations and functions in the structure of the 'Gym', I realized that if we go to the gym at 12:00, we can all complete our workouts and leave by 1:20. Moreover, following the "goesOnAfter" function and "fightsWith" relation, we have the following structure satisfying our conditions:

*Note: The colors represent the times each person gets on a machine. We see that this model follows "goesOnAfter", "timeLength", and ensures that no one should fight with anyone else.

Dumbbells (10 minutes)	Bike (20 minutes)		Treadmill (30 minutes)		
D	B1	B2	TM1	TM2	TM3
Lauren (12)	Katie (12)	Jo (12)	Sarah (12)	Josh (12)	Tim (12)
Me (12:10)	Lauren (12:20)	Laural (12:20)	Me (12:30)	Peter (12:30)	Katie (12:30)
Peter (12:20)	Sarah (12:40)	Nikki (12:40)			
Josh (12:30)	Josh (1)				
Jo (12:40)					
Noah (12:50)					
Tim (1)					
Claire (1:10)					

Times and Groups in Common Grounds (smallPool, cafeteria)

smallPool	cafeteria
12: Claire, Noah, Me	12: Peter, Nikki, Laural
12:10: Claire, Noah, Lauren	12:10: Peter, Nikki, Laural
12:20: Claire, Noah, Katie	12:20: Me, Nikki, Jo
12:30: Claire, Noah, Tim	12:30: Sarah, Nikki, Jo
12:40: Claire, Noah, Tim	12:40: Lauren, Laural, Josh
12:50: Claire, Jo, Tim	12:50: Lauren, Laural, Josh
1: Claire, Jo, Noah, Me, Katie	1: Lauren, Laural, Sarah, Peter, Nikki
1:10: Tim, Jo, Noah, Me, Katie	1:10: Lauren, Laural, Sarah, Peter, Nikki

1:20: Tim, Jo, Noah, Me, Katie, Lauren, Laural, Sarah, Peter, Nikki, Josh, and Claire are finished and ready to leave.