

## Report for Lab 1

Atreya Misra  
am73676

### Part A:

To prove that there is always stable matching, the first thing that is important to understand is that no tenant can be rejected by all landlords. A landlord can only reject a tenant if their apartment is taken. So if all landlords for apartments reject a tenant, all the apartments must be full. However, since there are an equal number of apartments as tenants, that scenario is impossible. No tenant can occupy two apartments and vice versa. In addition, a tenant and an apartment are always paired. Furthermore, since there is a pairing every time, no tenant tries to pair with the same apartment twice. If a landlord rejects a tenant, which must mean that he has a tenant that he already prefers. Also, if the landlord swaps pairing with another tenant, the landlord will always have his most preferred tenant of the tenants that requested an apartment under the landlord. Hence, a stable matching will always be present because of all the pairings available to a landlord; he will pick his best choice.

### Part B:

The first thing my algorithm does is makes an arraylist of arraylists for apartment preferences. It takes what the landlord prefers, and assigns the preferences to the apartment they own. After this occurs, a tenant that has not yet found an apartment is going to pair with an apartment that it prefers that is already free. If the tenant's preferred apartment is not available, we check the apartment's preference over tenants to see whether the landlord that owns the apartment prefers the tenant that currently wants it. If this is true, then a new pairing will be made with the tenant and the apartment and the current tenant paired with the apartment will be freed.

### Part C:

Some of this proof will be taken from part A. Since the algorithm goes step by step to ensure that every tenant finds an apartment, then every tenant and apartment will be filled. To show that it is stable matching, every tenant the wants an apartment will either be accepted or rejected by a landlord. If a landlord accepts a tenant, which means the landlord did not have the apartment that the tenant wants filled or the landlord prefers the tenant to the tenant already in the apartment. If the tenant is rejected, that means the landlord's apartment is filled with a better or equally qualified tenant. In addition, since a tenant is going down his or her list of preferred apartments until there is a match and keeps going until the end of the algorithm, every tenant will have the best option possible.

Part D: My algorithm is  $O(n^2)$  because there are  $n$  tenants that examine  $n-1$  apartments for each tenant.

Part E: Brute Force is  $O(n^4)$  because you are iterating  $n*n*n*n$  times for the brute force solution to get every possible case.

Part F:

