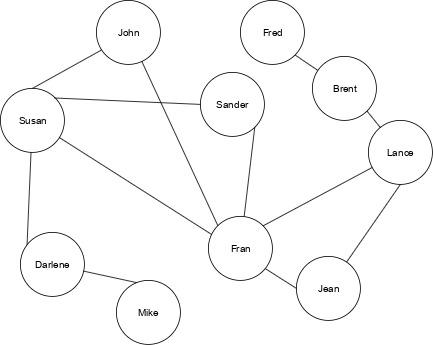
Jonathan Reznik, ID#1654919

CS20B HW6

1. Hashing
   1. not a good hashing algorithm that will create too many collisions….because age is not unique enough
   2. this could work, but depends largely on what the domain or population is e.g. with many people of the same last name, there could problems e.g. mapping members of the largest mafia families in NYC
   3. this seems to works the full name of the person is a good possible hashing algorithm
   4. not sure about this one...it does also seem fine, but what if any point to adding a constant number to the end of a hash code I don’t know because the compression algorithm will somehow be applied anyway
2. The code seems ok because the map relationships calls method get with argument k1 which is supposed to be a key object. The point of the get function is to check if the key exists in which case it will return the value and it will of course be != null. Therefore the code here in this case seems to work fine and I wouldn’t rewrite the code in any way, unless perhaps something else is wrong with it that I’m missing here.



* 1. “works with” because the edge is undirected meaning it has to apply both ways
  2. Unweighted graph has just boolean values in it for the adjacency matrix

|  | John | Susan | Fred | Sander | Fran | Mike | Lance | Jean | Brent | Darlene |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| John | N/A | T |  |  | T |  |  |  |  |  |
| Susan | T | N/A |  | T | T |  |  |  |  | T |
| Fred |  |  | N/A |  |  |  |  |  | T |  |
| Sander |  | T |  | N/A | T |  |  |  |  |  |
| Fran | T | T |  | T | N/A |  | T | T |  |  |
| Mike |  |  |  |  |  | N/A |  |  |  | T |
| Lance |  |  |  |  | T |  | N/A | T | T |  |
| Jean |  |  |  |  | T |  | T | N/A |  |  |
| Brent |  |  | T |  |  |  | T |  | N/A |  |
| Darlene |  | T |  |  |  | T |  |  |  | N/A |