**Q.10: (Planning a party).**

**Alice wants to throw a party and is deciding whom to call. She has n (which is at least**

**11) people to choose from and she has made up a list of which pairs of these people**

**know each other. She wants to invite as many people as possible subject to the**

**following two constraints:**

**1. Every person invited should know at least five other people that are invited.**

**2. Every person invited should not know at least five other people that are invited.**

**Design an efficient algorithm for maximizing the number of people she can invite.**

**Remember to analyze the running time and correctness.**

**Hint: Maximizing the number of invitees is the same as minimizing the number of**

**people Alice doesn’t invite. Obviously Alice might not be able to invite everyone. For**

**example, if one of the n people knows less than five people out of the n potential**

**invitees then the first constraint can never be satisfied for that person.**

Solution: Label the people that could be invited 1, . . . n. For a subset S of {1, 2, . . . , n}, define Ki(S) to be the number of people in S that the ith person knows and define Di(S) be the number of people in S that the ith person doesn’t know. Then the algorithm i

• Let P = {1, 2, . . . , n} be the set of potential invitees.

• While there exists i ∈ P such that Ki(P) < 5 or Si(P) < 5: P ← P \ {i}.

• Return P.

Correctness: For all people in the final set P (which could be empty), there are at least 5 people they don’t know in P and 5 people they do know in P. Hence, inviting all people in P satisfies Alice’s constraints. Suppose the optimal set of invitees is Popt. We use induction on the number of people we remove to prove P always contains Popt. Initially, when we have removed 0 people, P contains Popt. Assume after removing k people, P contains Popt. If we remove a (k + 1)th person then this person can not have been in Popt since if Ki(P) < 5 or Di(P) < 5 then Ki(Popt) < 5 or Di(Popt) < 5. Hence Popt is still contained in P. The running time is O(n 2 ). There are O(n) iterations and in each we need to scan through the |P| ≤ n remaining possible invitees to find if there is an i such that Ki(P) < 5 or Si(P) < 5. If so, we need to update |P| − 1 values of Kj and Sj .