* The Hypercube Algorithm: is a hash-based algorithm. It is parametrized by a vector of shares. Each server is mapped to a distinct point in the k-dimensional hypercube.
* The HC algorithm distributes relation Rj only to the subcube with dimensions corresponding to the variables in Rj. If the database is skewed, then, for any choice of the hash functions, at least one server will exceed this load.
* Get the load of the Hypercube algorithm using the equation in terms of the shares, we can compute the optimal values of the shares to minimize the load using Linear Program.
* For uniform database, (N1 = N2 = ··· = N¸ = Nl), the optimal shares and optimal load of the HC algorithms can be expressed in terms of v\* and t\* as follows: for all i in [k] : pi =p ^(vi\*/t\*), L = N/p^(1/t\*)
* For non-uniform database, the cardinalities of the input relations are not equal, the algorithm takes advantage of the unequal sizes of the input relations, e.g. by allocating more shares to the larger relations, and can achieve better speedup than ~ p^(1/t\*)
* Computing a query in one round or multiple rounds of communication: multiple rounds reduce the load for each processor but incurs a significant cost. Nevertheless, there are cases where running a query in multiple rounds can significantly reduce the communication cost per round, and sometimes it makes sense to trade off rounds for communication per round.