Grover's Quantum Search Algorithm

For solving Real-Life Satisfiability Problems

Introduction

- An algorithm proposed by Lov Grover to solve the problem of an unstructured search
- Logically, Quantum algorithm for finding the input value w of a function f(x) with f(w)=1 and f(x)=0 for all other values of x
- Example of a problem: Finding a phone number in an unsorted telephone directory
- Why Grover's Algorithm? : To find the marked item using classical computation, you need to check an average of boxes N/2, or worst case, all of them. But with a quantum computer, Grover's algorithm can find the marked item in roughly $\sqrt{N}$ steps

Satisfiability Problems

- The satisfiability problem asks the computer to find a set of values (commonly true or false) for several variables such that they satisfy certain constraints.

  Examples:

- Sudoku Problem::

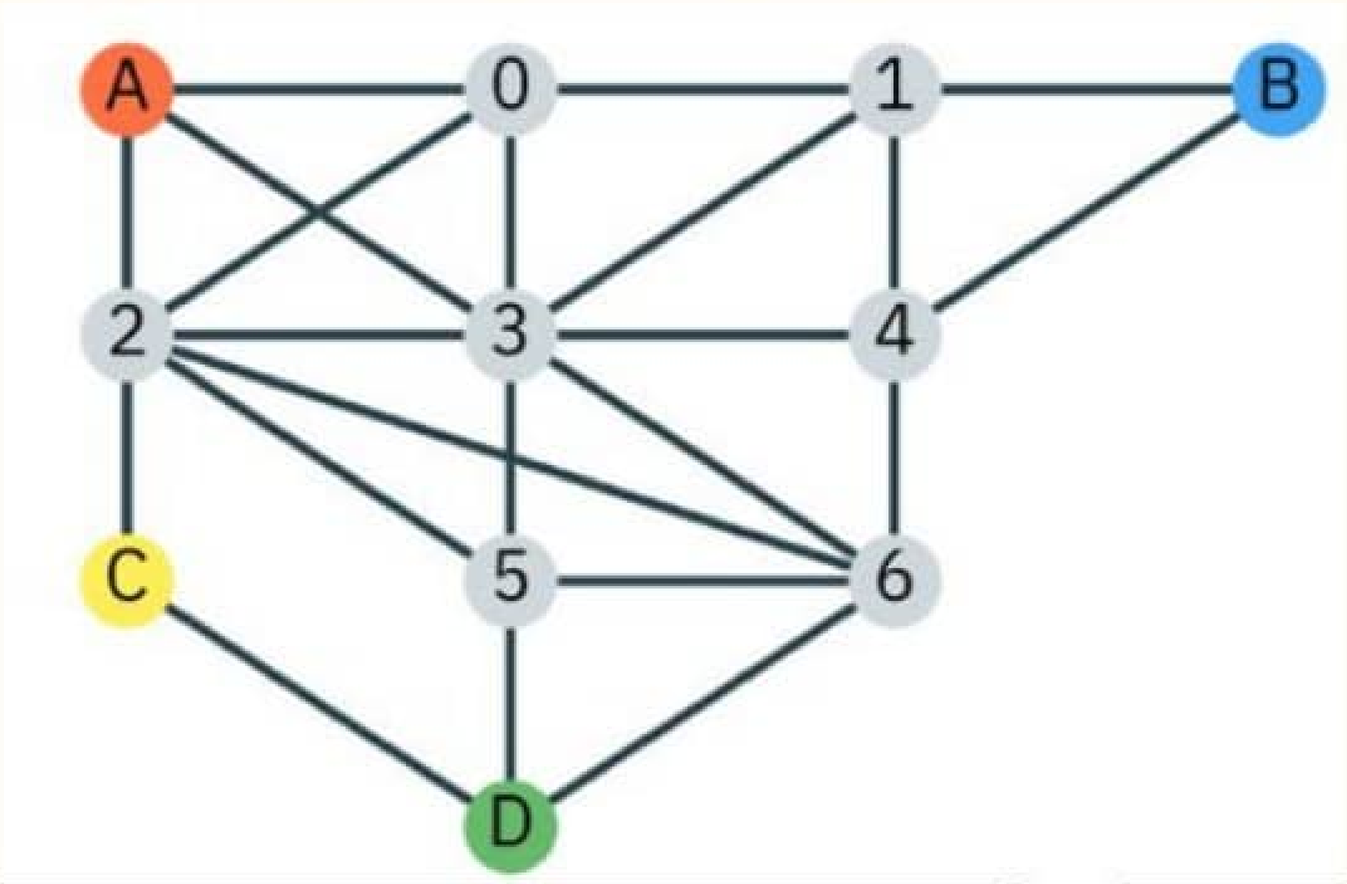
  Problem: Given a partially filled 9x9 grid, fill in the remaining empty cells with digits from 1 to 9, such that each digit appears only once in each row, column, and 3x3 subgrid.

- Graph Coloring Problem:

  Problem: Given an undirected graph, assign colors to the vertices in such a way that no two adjacent vertices have the same color, using the fewest number of colors possible

Problem Statement

Gandaki is a province in Western Nepal consisting of 11 districts. Four Departments of Incubate Nepal A, B, C and D have each established their first centres in this province in non-overlapping districts. The graph below shows the 11 districts of Gandaki Province and which district has an Incubate branch already.



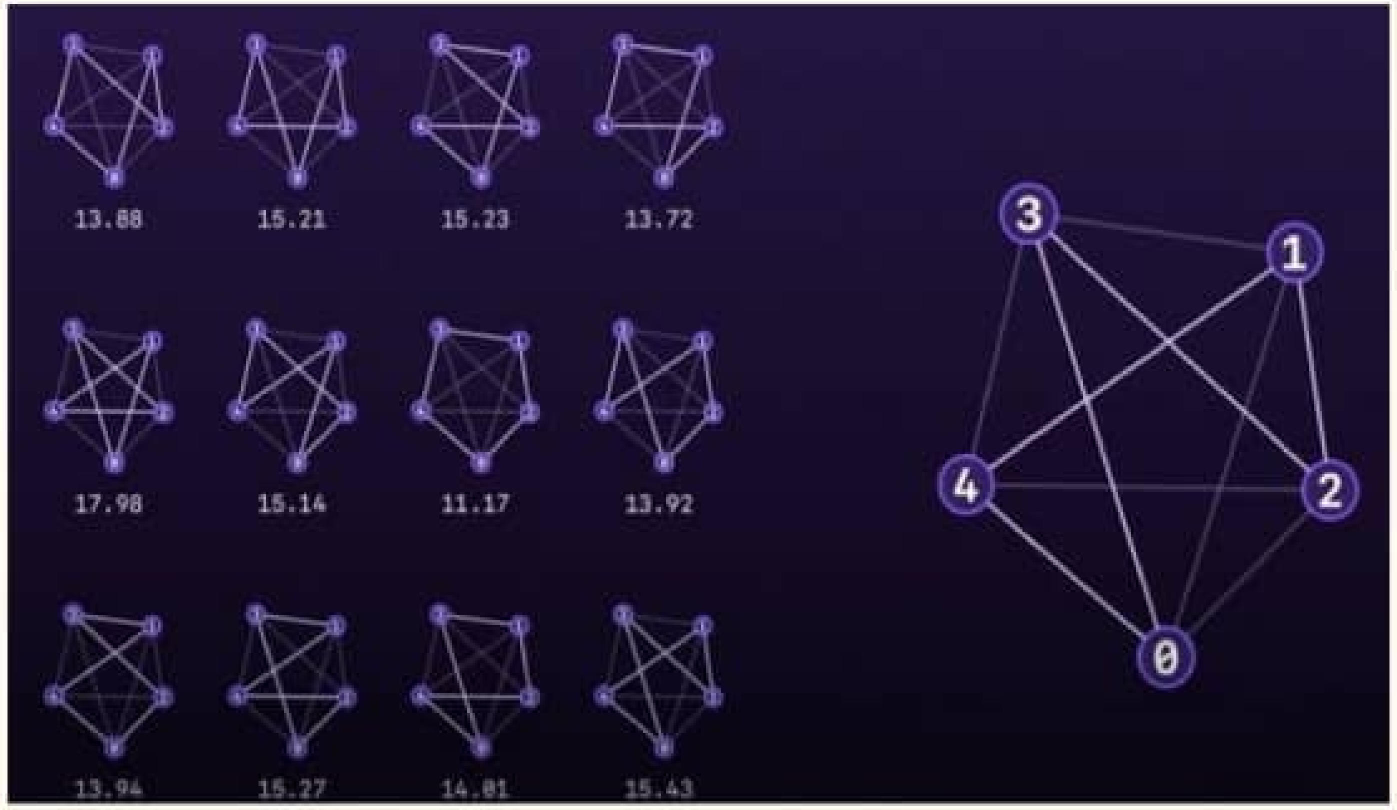
Chairman of the Province wants to establish Incubate branch in the rest of the districts that still don't have one yet. Upon his request, all four Incubate departments discussed with each other and agreed to establish their branches under the following two conditions: Only one department is allowed in one district and No two adjacent districts can have Incubate branch from the same department.

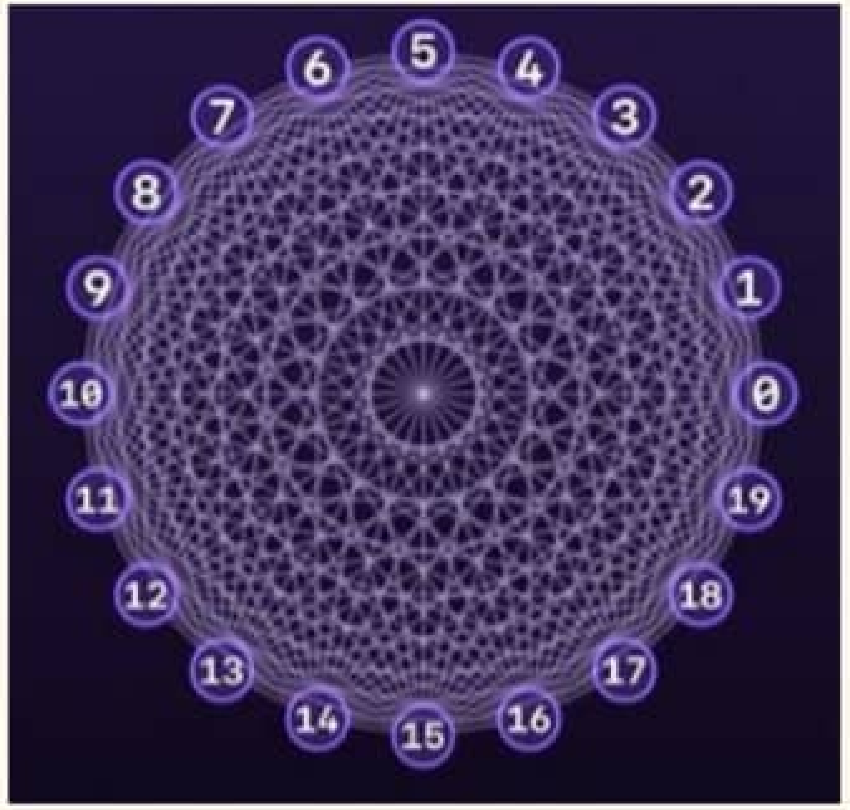
Aim: Can you create a plan that meets the given conditions? Please provide all possible combinations of plans that satisfy the conditions mentioned above

Classical Approach

Brute Force method: The most expensive and simplest classical solution to the problem is to find the solution by brute force method. However, the problem becomes impossible to solve when a large number of cities are taken.

For N cities, (N-1)! possible iterations are needed to search for the solution, which shoots up very fast as N increases.



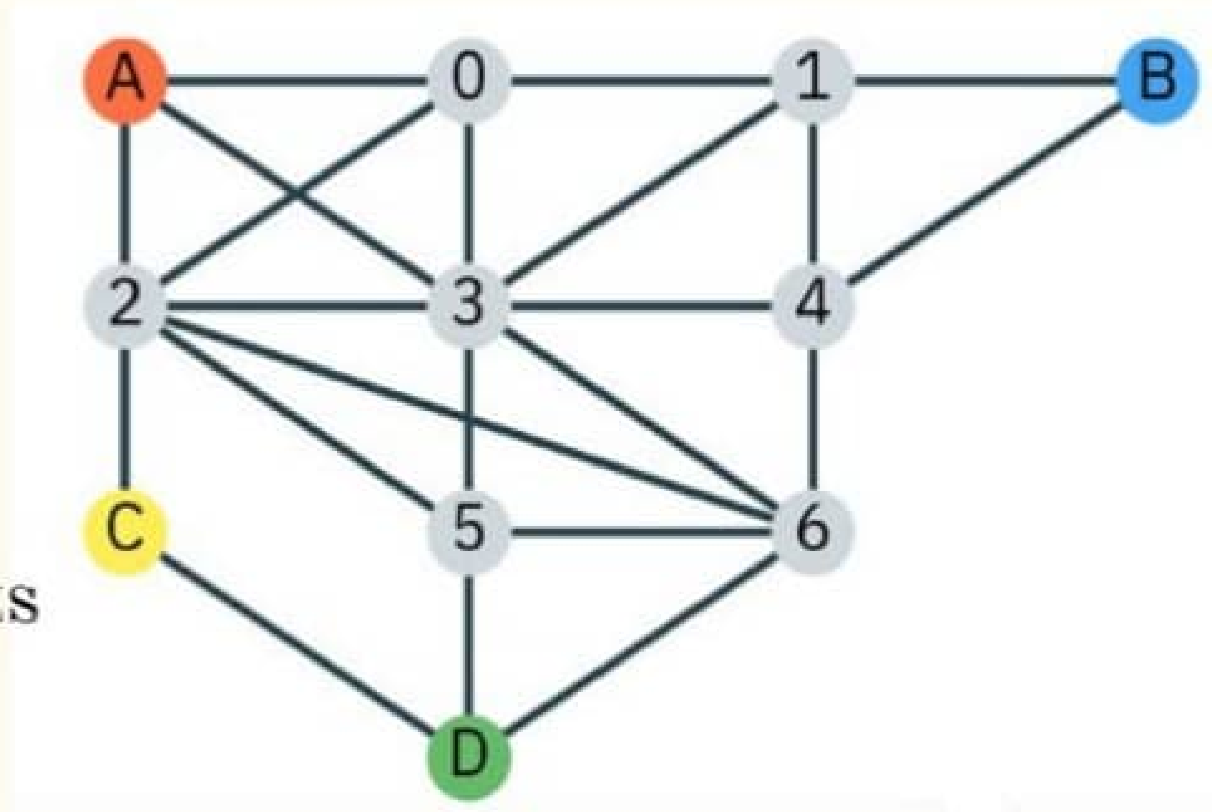


Grover's Algorithm Overview

We can solve the problem using Grover’s algorithm which can be summarized in the following steps:

- ❑ Step I: Create a superposition of inputs
- ❑ Step II: Build an oracle according to the constraints of the problem
- ❑ Step III: Diffusion
- ❑ Step IV: Measurement

Quantum Representation (Superposition of Inputs)



Assignment of input into quantum registers:

- Since each district has 4 possibilities(ABCD) which means 4 different states. So to store information about 7 empty districts we need 14 qubits(q[0]-q[13]). $\log_2 4$ per districts.

- Create a superposition of department across q[0]~q[13] for all possible combinations where A:00, B:01, C:10, D:11

- **Optimization**: Narrowing down superposition of states in districts that already have department A, B, C or D adjacent to them.

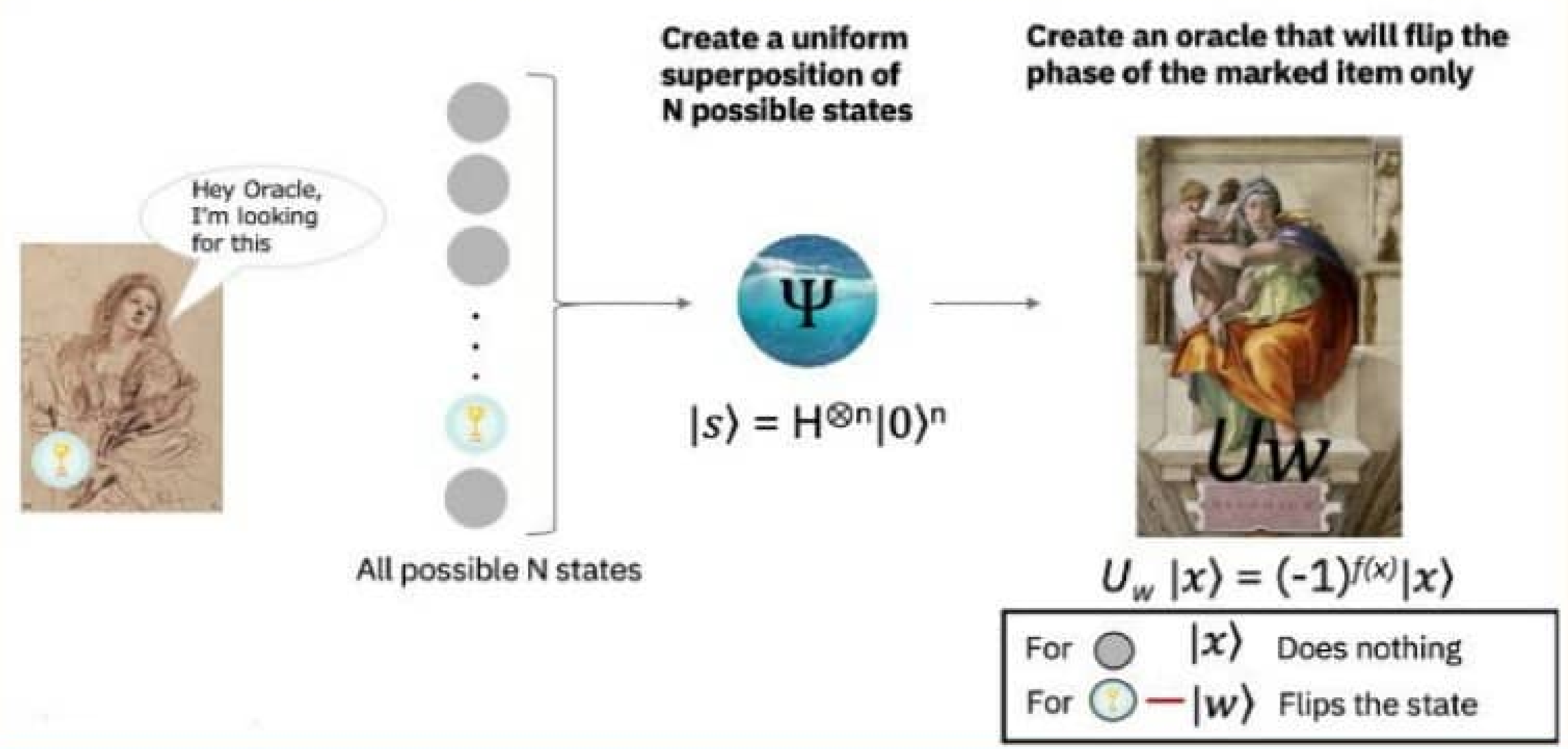
  - District 2 is neighboured by dept A and C so you can eliminate $|00\rangle$ and $|10\rangle$ and create a superposition of $|01\rangle$ and $|11\rangle$ only to initialize this 2.

Oracle

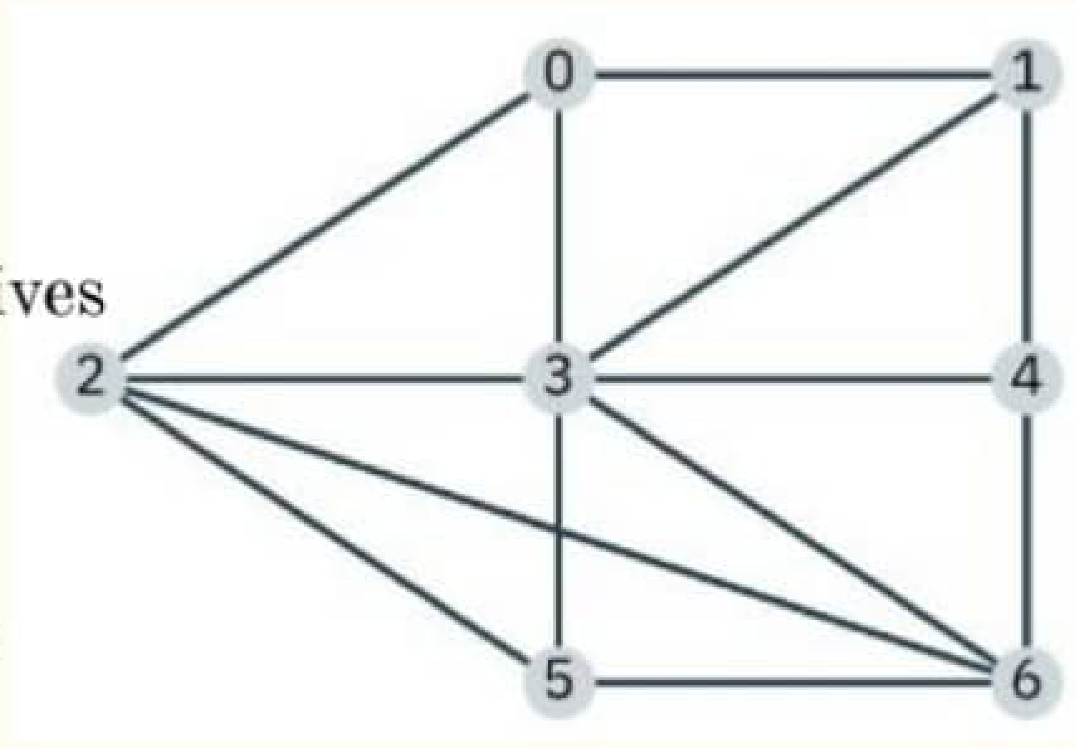
How will the list items be provided to the quantum computer? A common way to encode such a list is in terms of a function *f* that returns f(x)=0 for all unmarked and f(w)=1 for winner.

An oracle refers to a specific type of quantum gate or operation that provides information or performs a specific task related to the problem being solved. The oracle in Grover's algorithm is designed to mark the desired solutions within the search space. It acts as a "black box" that identifies the solutions you are interested in finding. Typically, the oracle in Grover's algorithm is a unitary operation that flips the phase of the marked solutions, while leaving the other states unchanged.

Definition: Oracle matrix **Uw** act on any standard basis states $|x\rangle$ by $U_w|x\rangle = (-1)^{f(x)}|x\rangle$



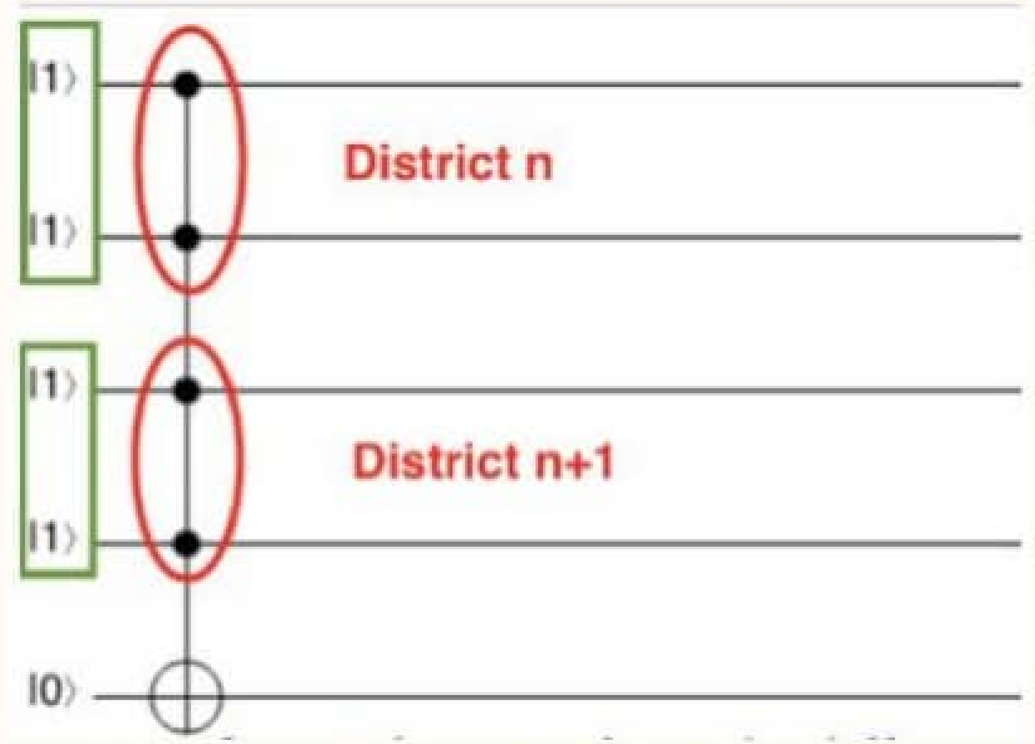
Oracle Construction

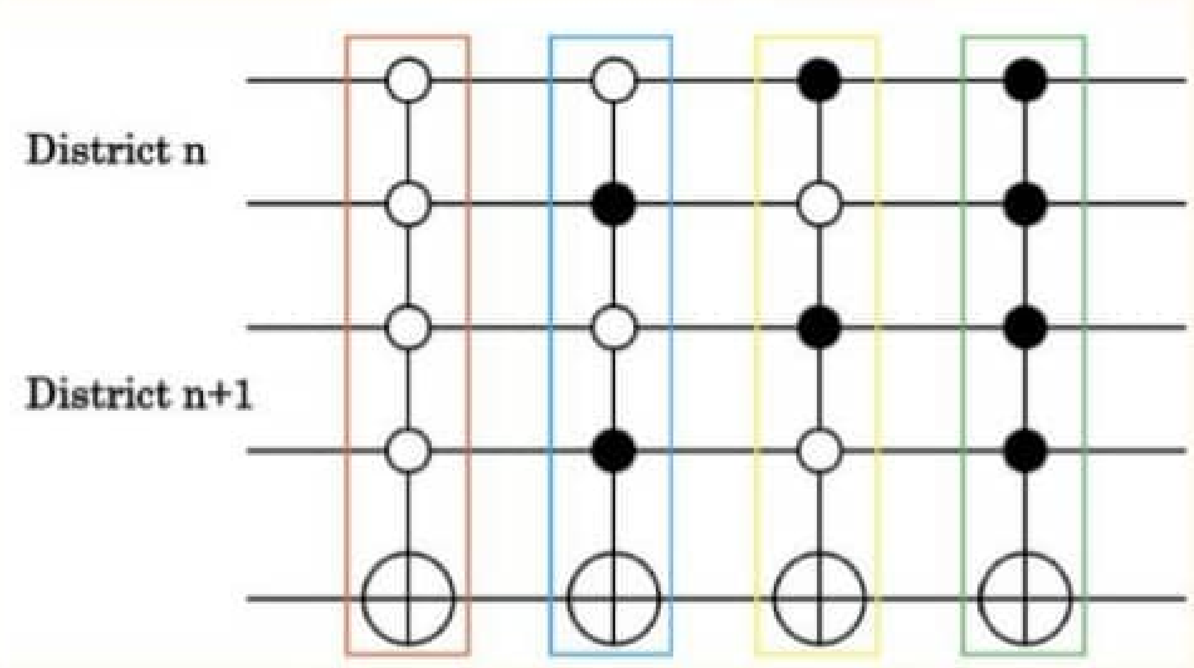


After optimization step 1, we can focus on the remaining districts which gives us a graph representation as follows with 13 edges. For data storage inside oracle we require 13 qubits.

Aim: Create an oracle that checks each neighboring pair of edges whether they have a different department or not.

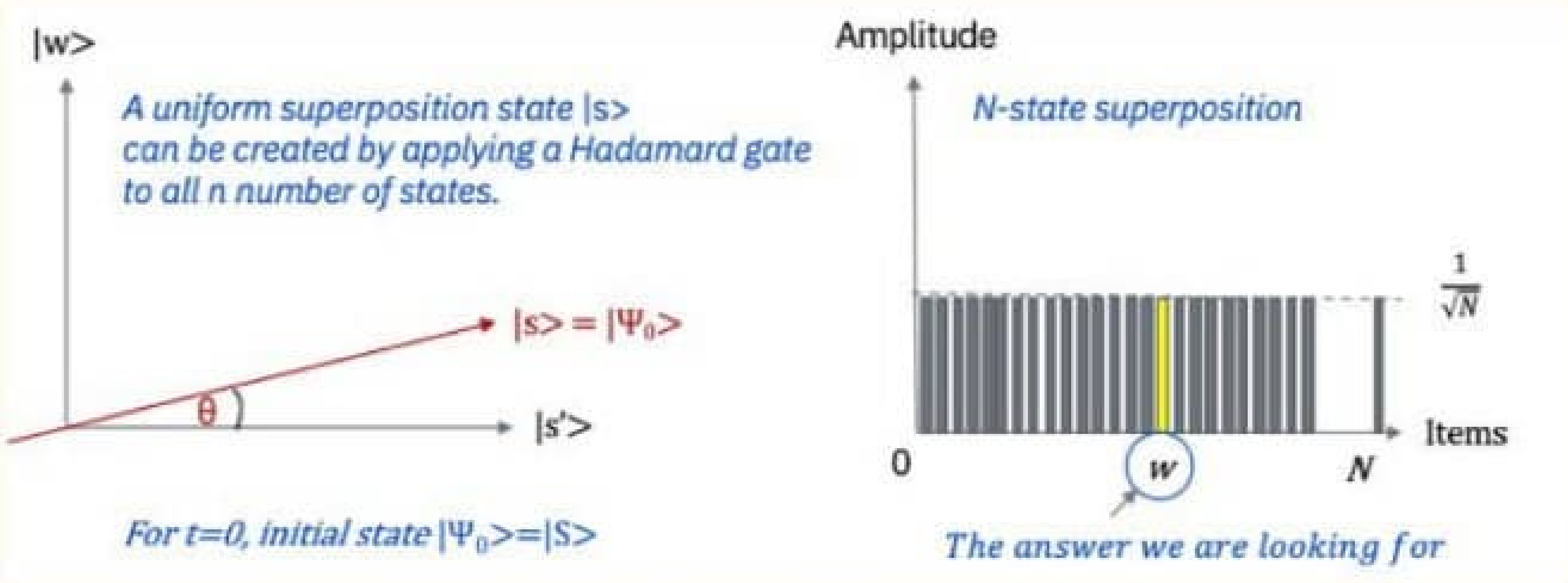
To do this, we can create a circuit to check if district n and its neighboring district n+1 has a different Incubate department or not.



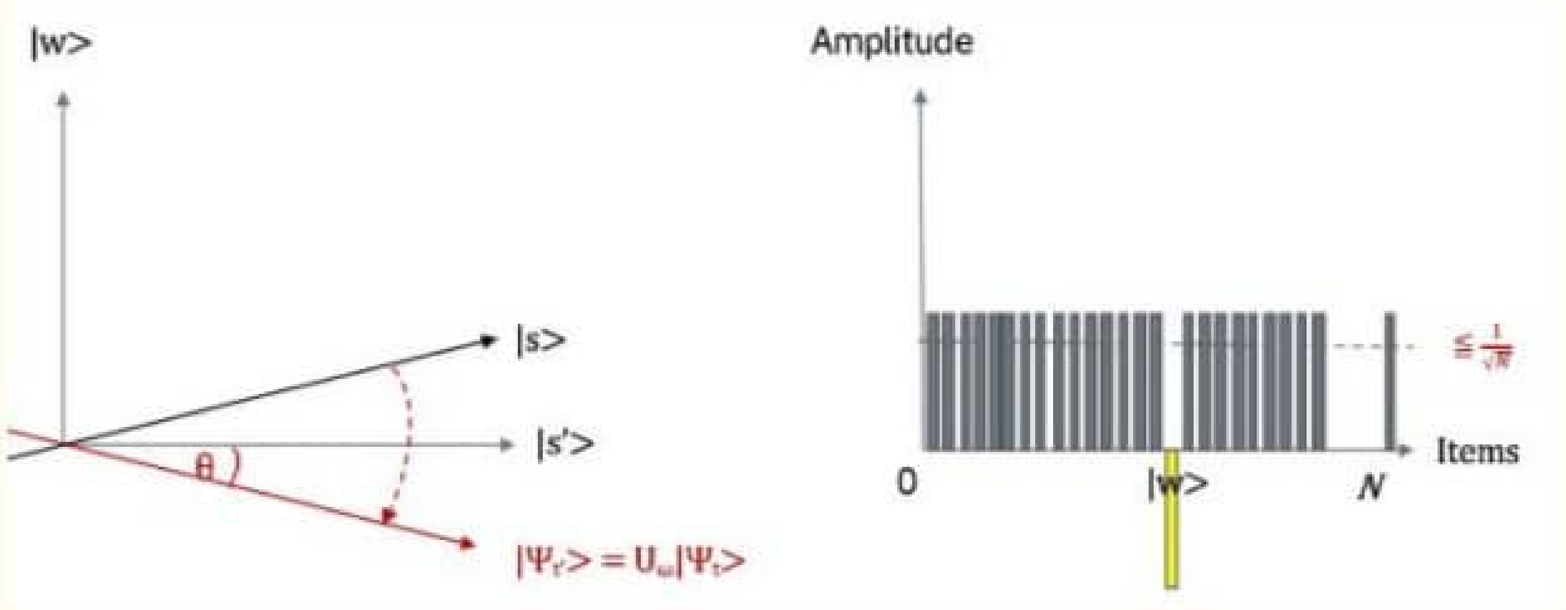


Diffusion

- Oracle Reflection Uw



Step 1: $|s> = H^{\otimes n}|0>^n$ $|s> = \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x>$



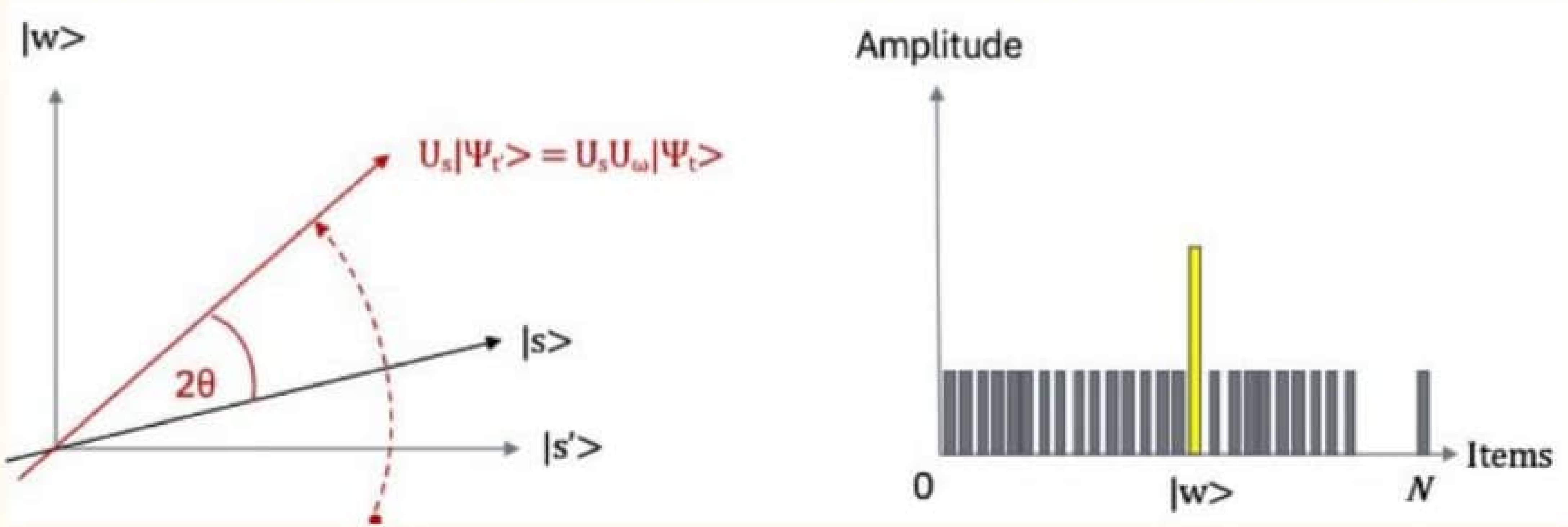
Step 2: Oracle reflection Uw to state $U_w|\psi_t> = |\psi_{t'}>.$

Create a diffusion circuit Us that reflects about initial uniform superposition | s>

Step 3: Apply additional reflection Us about the state | s> $U_s = 2|s><s| - I$

Note: UsUw rotates the initial state | s> closer towards the winner | w>. This is for general case.

In our case: $U_s = 2|s><s| - I = 2U_{const}|0><0|U^\dagger_{const} - I = -U_{const}(I - 2|0><0|)U^\dagger$ **s.t** $U^\dagger_{const}$ **is the inverse of Ucon**



Iterations and Runtime Analysis

Let no of iterations we require to get high probability of $|w\rangle$ be 'r' i.e. $(U_s \cdot U_w)^r |s\rangle \approx |w\rangle$.

We saw after 1 operation i.e. $U_s \cdot U_w \cdot |s\rangle$, the angle w.r.t. $|s'\rangle$ was $\theta + 2\theta = 3\theta$.

Our goal: $3\theta \sim \frac{\pi}{2}$ (when?)

Since after every iteration angle increase by 2θ, we can write this as,

$$r \cdot 2\theta + \theta = \frac{\pi}{2} \quad \text{where } \theta = \sin^{-1}\left(\frac{1}{\sqrt{2^n}}\right)$$

$\Rightarrow r = \frac{\pi}{4\theta} - \frac{1}{2}$ [Now, where 2^n is very large $\sim N$ then θ will be very small]

So, by small angle approx, $\sin\theta \approx \theta$.

$$\therefore r = \frac{\pi}{4 \cdot \frac{1}{\sqrt{2^n}}} - \frac{1}{2} \approx \frac{\pi}{4}\sqrt{2^n} \approx \text{order of } \sqrt{N} \quad O(\sqrt{N})$$

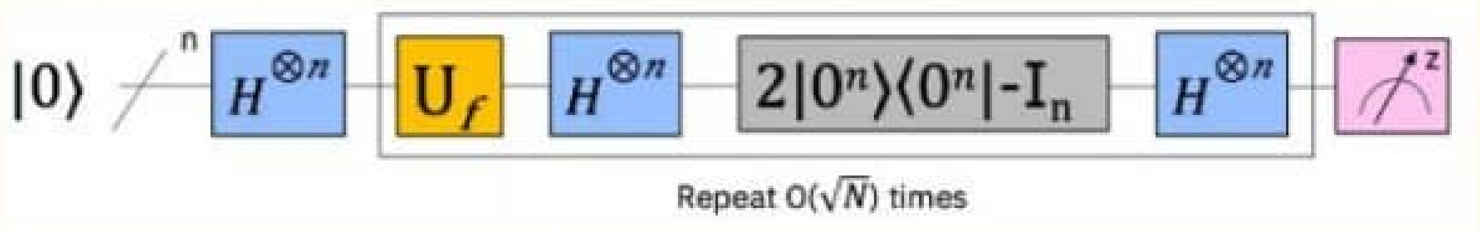
Why is $\theta = \sin^{-1}\left(\frac{1}{\sqrt{2^n}}\right)$

$$|s\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$$

$$|s\rangle = \frac{\sqrt{2^n - 1}}{\sqrt{2^n}} |s'\rangle + \frac{1}{\sqrt{2^n}} |w\rangle$$

$$= \cos\theta\, |s'\rangle + \sin\theta\, |w\rangle$$

$$\sin\theta = \frac{1}{\sqrt{2^n}} \quad \therefore \theta = \sin^{-1}\left(\frac{1}{\sqrt{2^n}}\right)$$



Measurement

We measure the qubits where the districts are mapped to

- Measurement score will grow large we can sort them and extract top results

- At first will get results in-terms of (0 and 1) in 14 bits with their highest probabilities `'00010111100001', 546`

- Will convert each 2 bits into their corresponding departments and add 4 other nodes for the districts which already have 4 departments `0201132203', 546`

Results and Conclusions