양자 프로그래밍 4 강
Grover’s Search Algorithm

Quantum Ant

HSU 한성대학교 HANSUNG UNIVERSITY

QUANTUM ANT
CRYPTO OF THE SWARM

CryptoCraft LAB

Grover’s Search Algorithm

- 정렬되지 않은 N개의 데이터에서 특정 데이터를 $\sqrt{N}$번 만에 높은 확률로 찾아내는 양자 알고리즘

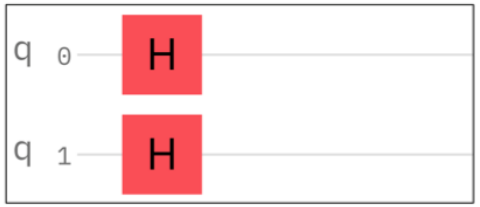
- Grover’s search : 크게 두 가지 단계로 구성
  - Oracle : N개의 데이터 중 **솔루션의 amplitude를 반전**시킴
  - Diffusion operator : Oracle에서 반환한 **솔루션의 amplitude를 증폭**시킴

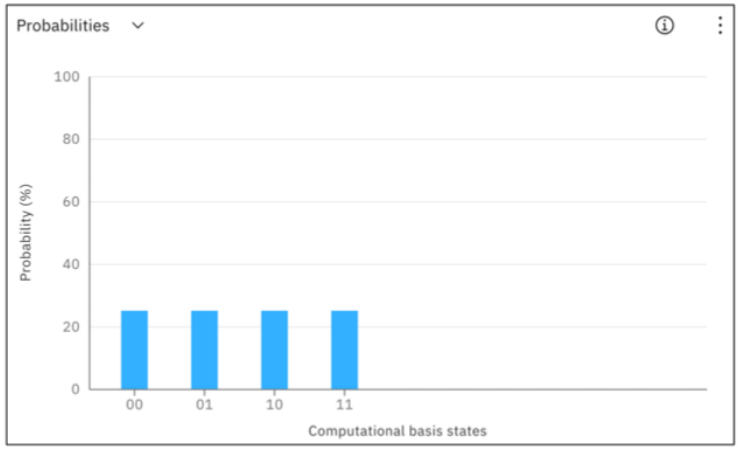
Grover's Search Algorithm : Setting

- 2-bit 즉, $N = 4$의 경우 Solution **01** 을 찾아내는 Grover search

- 2-qubit에 Hadamard 게이트를 적용하여 중첩 상태로 만듦 → 00, **01**, 10, 11이 모두 확률로서 동시에 존재하게 됨

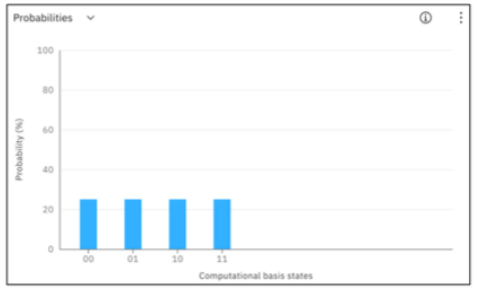
```
def Grover(eng):

    data = eng.allocate_qureg(2)

    All(H) | data
```

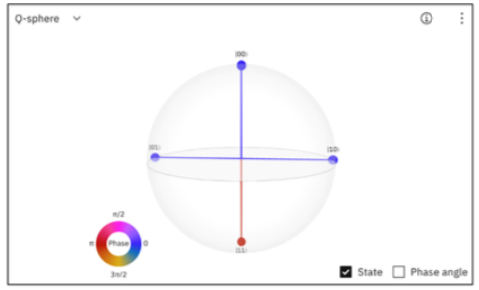


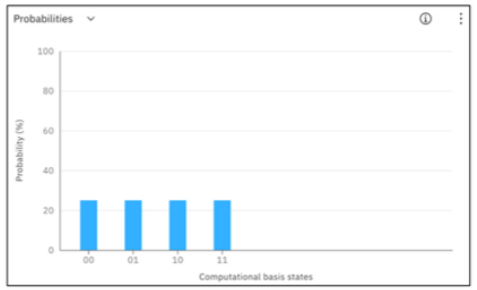


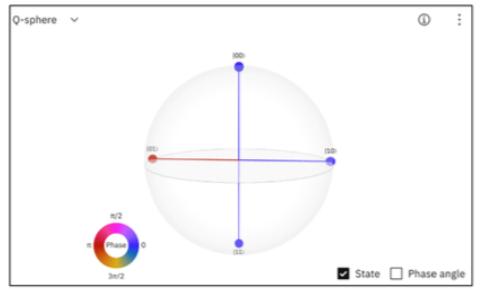
Grover's Search Algorithm : Design Oracle

- Solution **01** 을 찾아내기위한 Oracle을 설계해야 함
- 2-qubit 상태가 **01** 인 경우에만 2-qubit이 11인 상태를 가지도록 설계
  - 11, 즉 모든 큐비트가 1인 경우 Z게이트를 사용하여 해당 상태의 부호를 – 로 바꿈







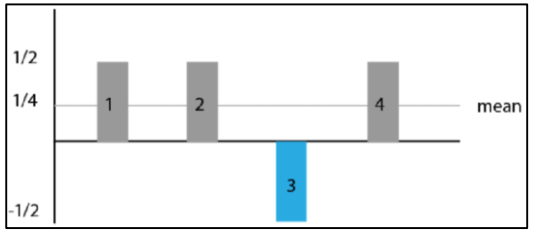


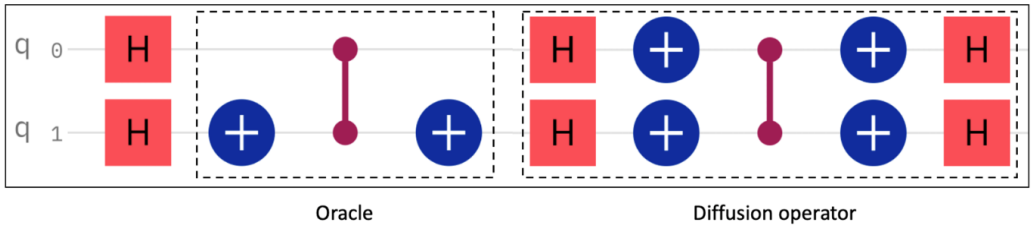

```
def Grover(eng):

    data = eng.allocate_qureg(2)

    All(H) | data

    # Oracle
    X | data[1]

    with Control(eng, data[0:-1]):
        Z | data[-1]

    X | data[1] # reverse
```

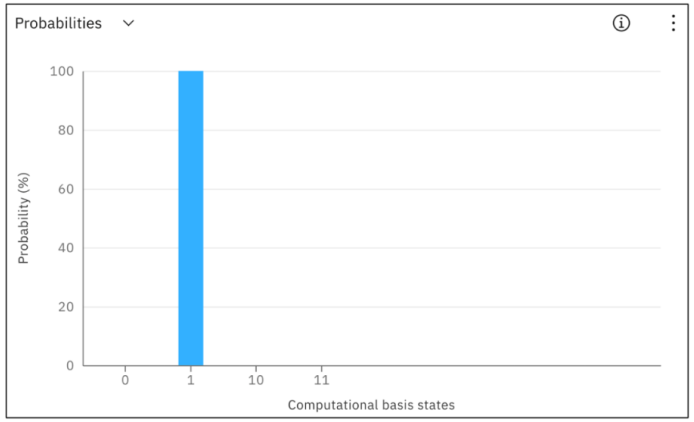
Grover's Search Algorithm : Diffusion operator

- Oracle에서 반환한(- 부호) 솔루션의 amplitude를 증폭시킴

평균 amplitude − (각 amplitude −평균 amplitude)







$\sqrt{N}$ 반복

```
def Grover(eng):

    data = eng.allocate_qureg(2)

    All(H) | data

    # Oracle
    X | data[1]

    with Control(eng, data[0:-1]):
        Z | data[-1]

    X | data[1] # reverse

    # diffusion operator
    with Compute(eng):

        All(H) | data
        All(X) | data

    with Control(eng, data[0:-1]):
        Z | data[-1]

    Uncompute(eng)


    All(Measure) | data

    print(int(data[1]), int(data[0]))
```

* $\lfloor \frac{\pi}{4} 2^{\frac{n}{2}} \rfloor$ (about $2^{\frac{n}{2}}$)

실 습

감사합니다