```
In[9]:= LazyQuantumRandomWalk[State0_, Steps0_] := Module[{State = State0, Steps = Steps0},
          BitOrder = 3;
```

$$\sigma = e^{\frac{\pi 2 i}{\text{BitOrder}}};$$

$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} = H = \frac{\begin{pmatrix} 1 & 1 & 1 \\ 1 & \sigma^{\text{BitOrder}-1} & \sigma \\ 1 & \sigma & \sigma^{\text{BitOrder}-1} \end{pmatrix}}{\sqrt{\text{BitOrder}}};$$

$$M = \begin{pmatrix}
a & 0 & 0 & 0 & b & 0 & 0 & 0 & 0 & 0 & 0 & c \\
d & 0 & 0 & 0 & e & 0 & 0 & 0 & 0 & 0 & 0 & f \\
g & 0 & 0 & 0 & h & 0 & 0 & 0 & 0 & 0 & 0 & i \\
0 & 0 & c & a & 0 & 0 & 0 & b & 0 & 0 & 0 & 0 \\
0 & 0 & f & d & 0 & 0 & 0 & e & 0 & 0 & 0 & 0 \\
0 & 0 & i & g & 0 & 0 & 0 & h & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & c & a & 0 & 0 & 0 & b & 0 \\
0 & 0 & 0 & 0 & 0 & f & d & 0 & 0 & 0 & e & 0 \\
0 & 0 & 0 & 0 & 0 & i & g & 0 & 0 & 0 & h & 0 \\
0 & b & 0 & 0 & 0 & 0 & 0 & 0 & c & a & 0 & 0 \\
0 & e & 0 & 0 & 0 & 0 & 0 & 0 & f & d & 0 & 0 \\
0 & h & 0 & 0 & 0 & 0 & 0 & 0 & i & g & 0 & 0
\end{pmatrix}^{\mathsf{T}};$$

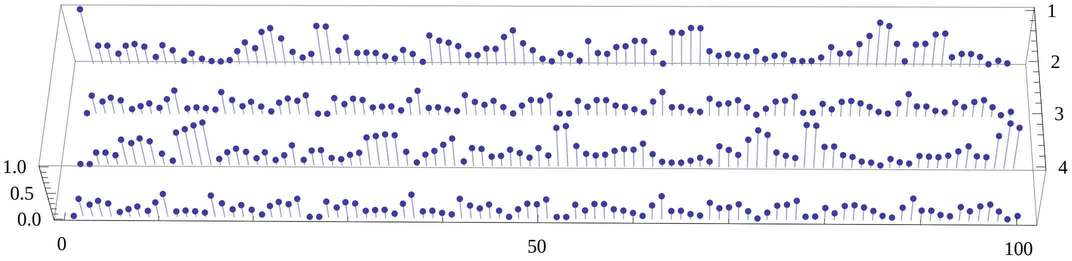
```
          For[j = 0, j < Steps, j++, State = Simplify[M.State]];
          Return[State];
        ]

In[10]:= FoutStepCircleStatesToPositionProbability[State0_] :=
          Module[{State = State0},
           ProbabilityAll = Simplify[Conjugate[State] * State];
           ProbabilaityMixed = Transpose[{{
               ProbabilityAll[[1, 1]] + ProbabilityAll[[2, 1]] + ProbabilityAll[[3, 1]],
               ProbabilityAll[[4, 1]] + ProbabilityAll[[5, 1]] + ProbabilityAll[[6, 1]],
               ProbabilityAll[[7, 1]] + ProbabilityAll[[8, 1]] + ProbabilityAll[[9, 1]],
               ProbabilityAll[[10, 1]] + ProbabilityAll[[11, 1]] + ProbabilityAll[[12, 1]]
              }}];
```

$$\text{For}\left[k = 1,\ k \le \frac{\text{Dimensions[ProbabilityAll]}[\![1]\!]}{\text{BitOrder}},\ k{+}{+},\ \sum_{j=(\text{BitOrder}\,(k-1))+1}^{\text{BitOrder}\,k} \text{ProbabilityAll}[\![j, 1]\!]\right];$$

```
           Return[ProbabilaityMixed];
          ]

In[11]:= InputState = SparseArray[{
            {12, 1} → 0,
            {1, 1} → 1
           }];

In[58]:= OutState = LazyQuantumRandomWalk[InputState, 1];

In[57]:= PositionProbability = FoutStepCircleStatesToPositionProbability[OutState];

In[51]:=
         WalkResult = {};
         For[LoopVar = 0, LoopVar < 100, LoopVar++,
          OutState = LazyQuantumRandomWalk[InputState, LoopVar];
          AppendTo[WalkResult, Flatten[FoutStepCircleStatesToPositionProbability[OutState]]]
         ]
```

In[56]:=
```
ListPointPlot3D[WalkResult, Filling → Bottom, BoxRatios → {2, 10, 1}]
```

Out[56]=


1
2
3
4
1.0
0.5
0.0
0
50
100