



SOLVING MAX CUT VIA GROVER

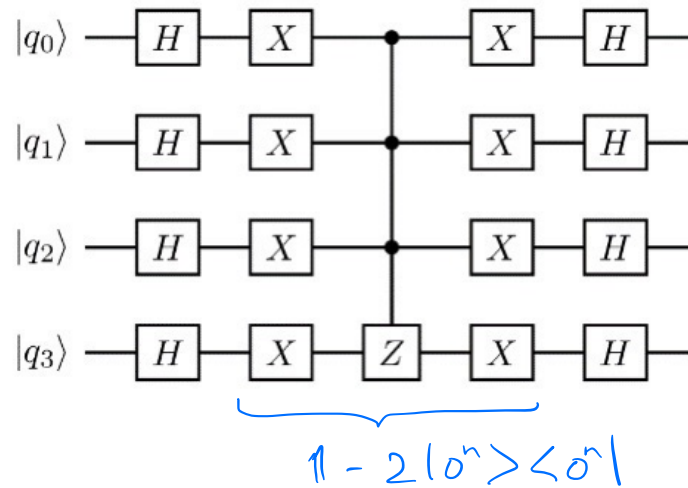
Jibran Rashid

Implementing the Diffusion Operator

$$1 - 2|h\rangle\langle h|$$

$$H^{\otimes n} (1 - 2|o^n\rangle\langle o^n|) H^{\otimes n}$$

$$Z_o = 1 - 2|o^n\rangle\langle o^n|$$



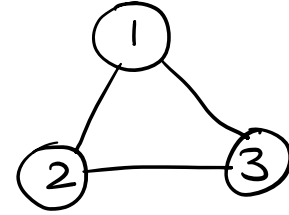
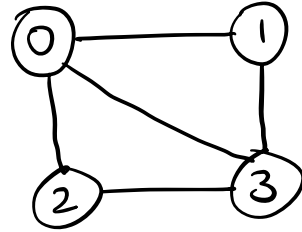
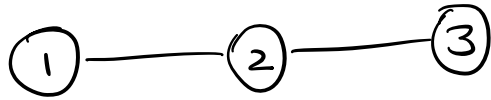
$$Z_o |x\rangle = \begin{cases} -|x\rangle & x = o^n \\ |x\rangle & \text{otherwise} \end{cases}$$

$$Z|0\rangle = |0\rangle$$

$$Z|1\rangle = -|1\rangle$$

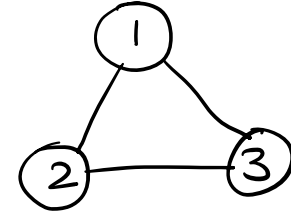
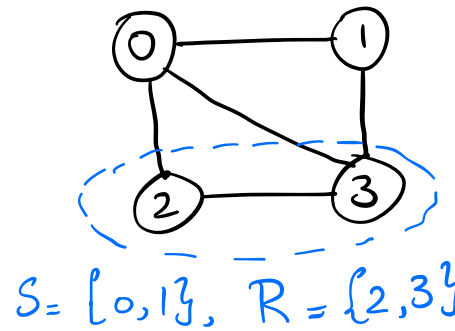
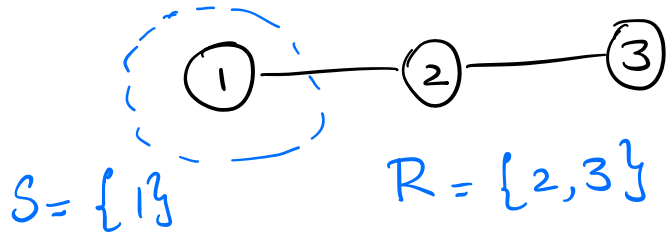
Undirected Graph

A pair $G = (V, E)$ where $V = \{v_1, \dots, v_n\}$ are the vertices & E contains unordered pairs of vertices, i.e., edges. $E = \{(v_i, v_j) \mid v_i, v_j \in V\}$



Undirected Graph

A pair $G = (V, E)$ where $V = \{v_1, \dots, v_n\}$ are the vertices & E contains unordered pairs of vertices, i.e., edges. $E = \{(v_i, v_j) \mid v_i, v_j \in V\}$

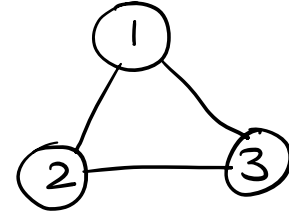
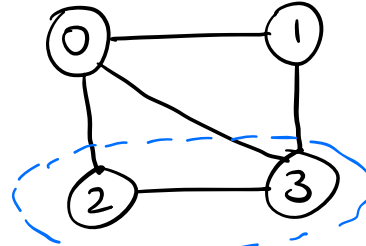
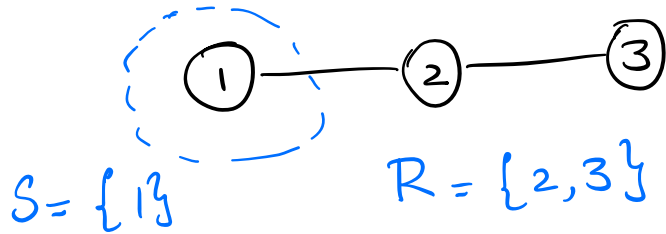


Cut

A partition of the vertices into two non-empty sets S & $R = V - S$

Undirected Graph

A pair $G = (V, E)$ where $V = \{v_1, \dots, v_n\}$ are the vertices & E contains unordered pairs of vertices, i.e., edges. $E = \{(v_i, v_j) \mid v_i, v_j \in V\}$



Cut

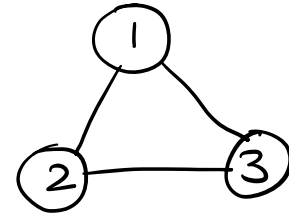
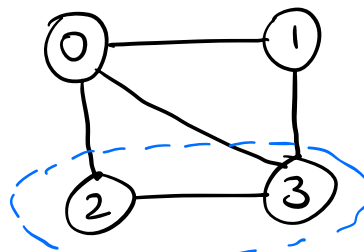
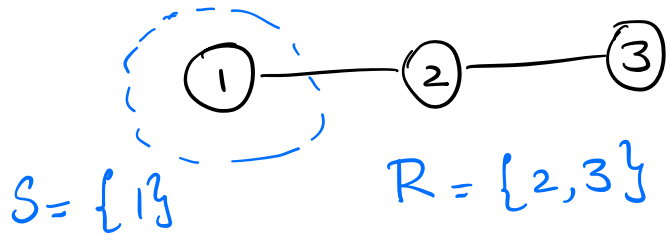
A partition of the vertices into two non-empty sets S & $R = V - S$

Size of a Cut

Number of edges b/w S & R .

Undirected Graph

A pair $G = (V, E)$ where $V = \{v_1, \dots, v_n\}$ are the vertices & E contains unordered pairs of vertices, i.e., edges. $E = \{(v_i, v_j) \mid v_i, v_j \in V\}$



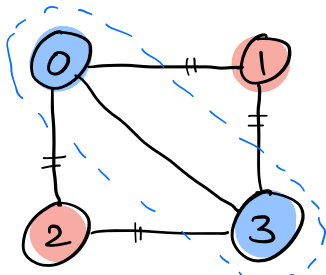
Cut

A partition of the vertices into two non-empty sets S & $R = V - S$

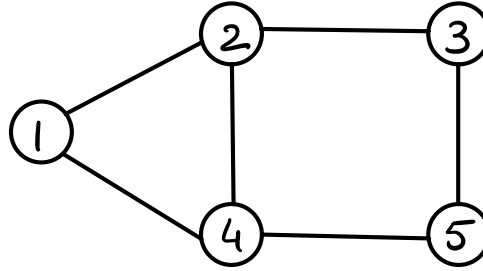
Size of a Cut

Number of edges b/w S & R .

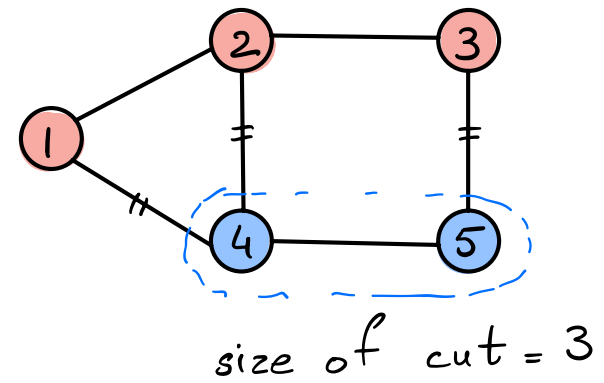
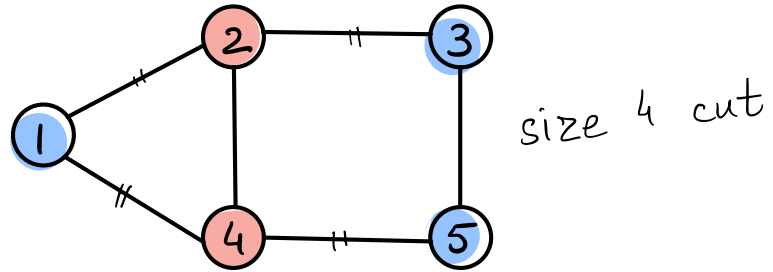
Max-Cut — Find partitions that maximize the number of edges across S & R .



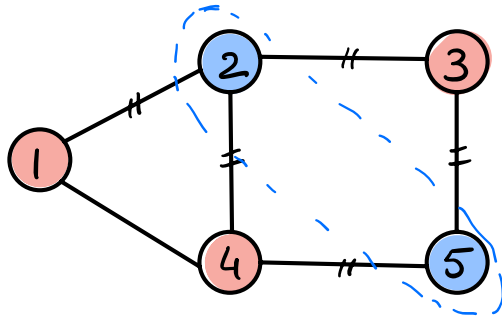
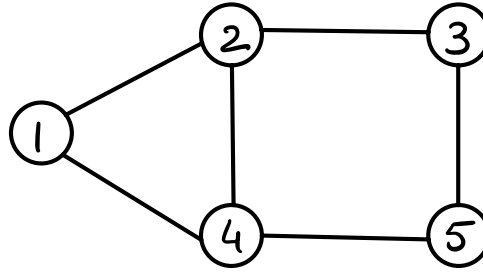
Max-Cut Example



Max-Cut Example



Max-Cut Example



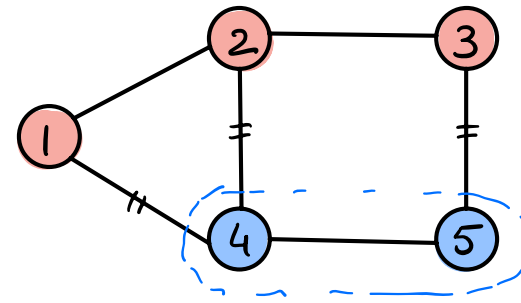
size of cut = 5

$$R = \{1, 3, 4\}$$

$$S = \{2, 5\}$$

$$f(1) = f(3) = f(4) = 0$$

$$f(2) = f(5) = 1$$



size of cut = 3

Decision Problem

Given G , does there exist a cut of size at least k .

— Decision vs. Optimization — Logarithmic overhead from Binary Search.

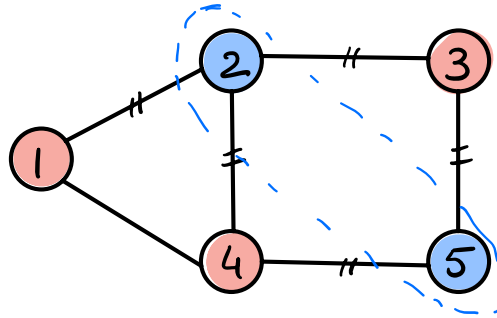
— Problem is known to be NP-Complete.

2^n possible partitions/colorings.

— Generic Quantum speedup via Grover to get $O(\sqrt{2^n})$.

Note: Min-Cut $\in P$ via maxflow algorithm

Max-Cut Example



size of cut = 5

$$R = \{1, 3, 4\}$$

$$S = \{2, 5\}$$

$$f(1) = f(3) = f(4) = 0$$

$$f(2) = f(5) = 1$$

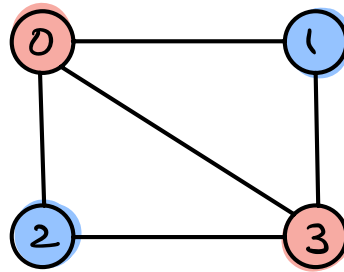
How do we construct the oracle?

Representing Vertices

n vertices $\longrightarrow$ n qubits

$|0\rangle \longrightarrow$ colored Red

$|1\rangle \longrightarrow$ colored Blue

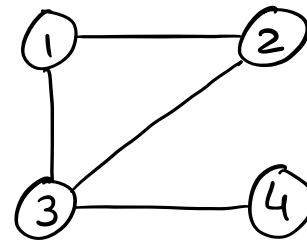
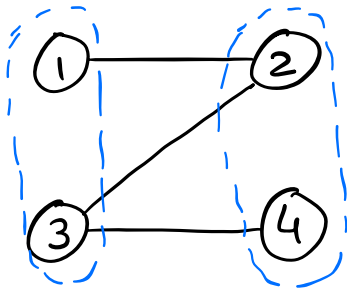


$$|v_0 v_1 v_2 v_3\rangle = |0110\rangle$$

Special Case: Bipartite Graphs

V can be partitioned/colored into two disjoint sets S & R , such that no edge exist between vertices in S & no edge exists b/w vertices in R .

Example



Not bipartite

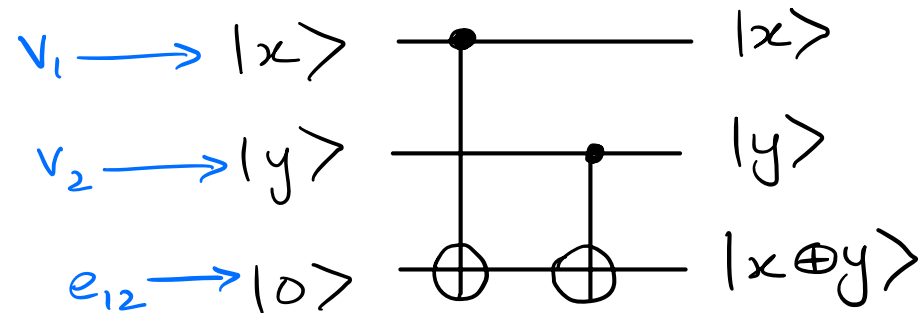
Bipartite graphs admit two coloring. (One color for each set)

So, max-cut solution is just total # of edges.

(Since all edges are cross-connecting)

Edge Checking

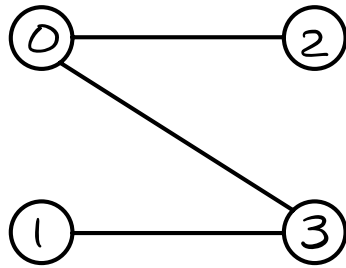
Does each end of an edge have vertices with different colors?



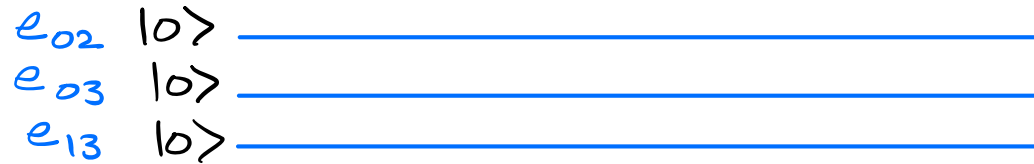
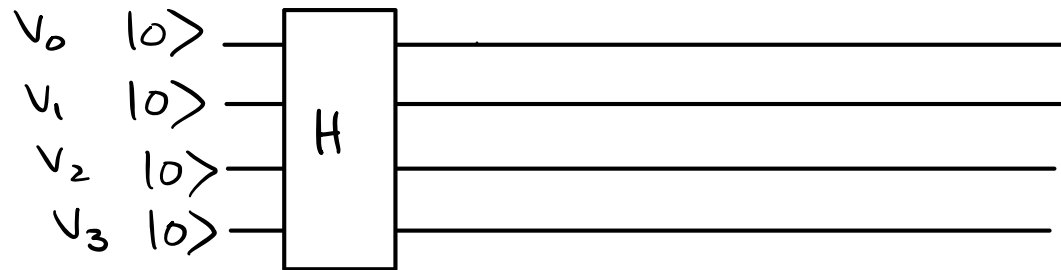
x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

} Answer to whether both vertices have different colors

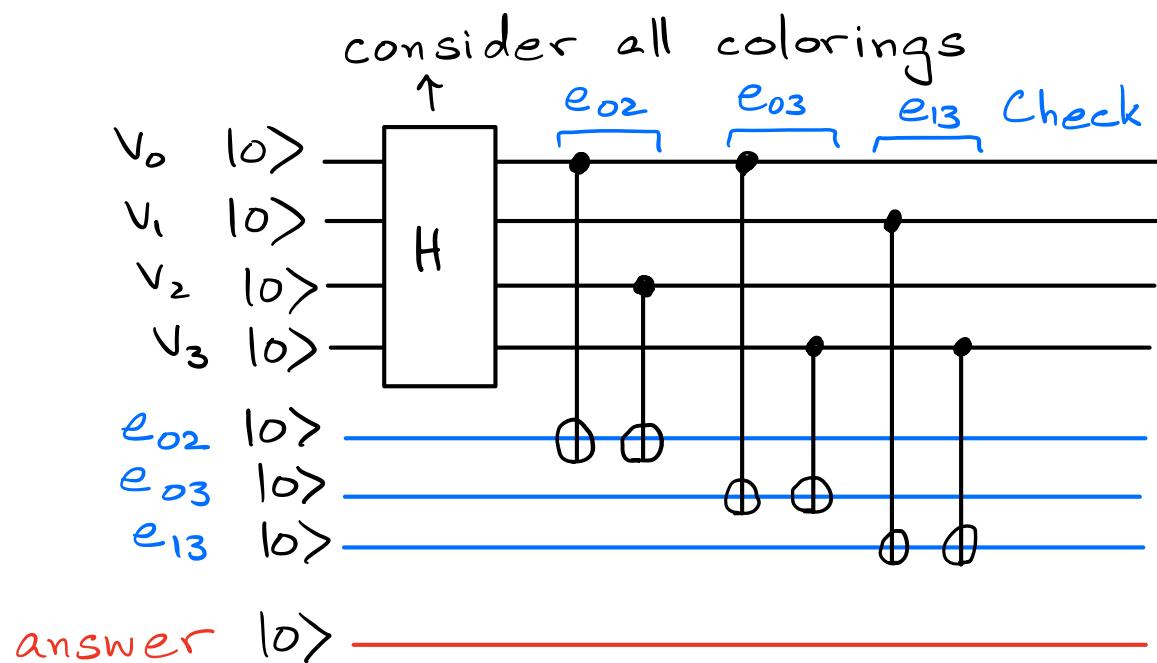
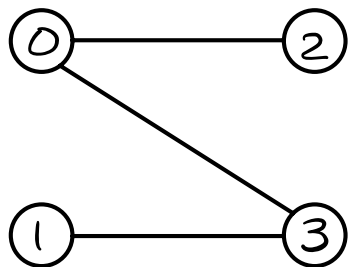
Oracle Construction



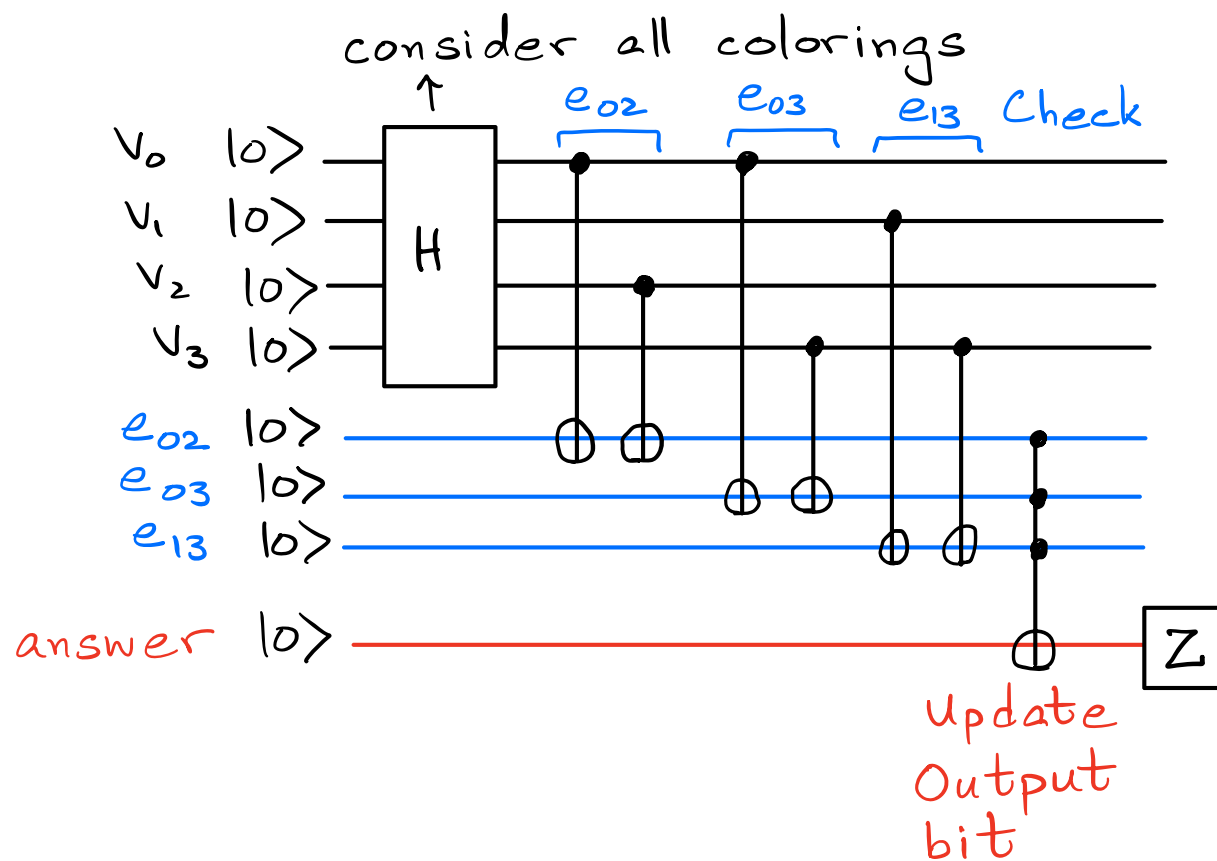
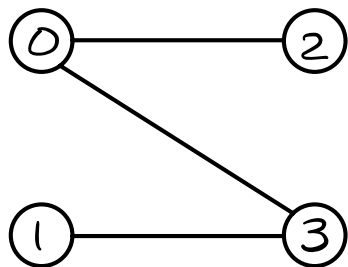
consider all colorings

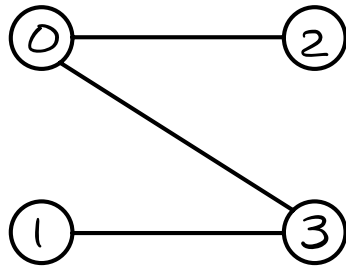


Oracle Construction

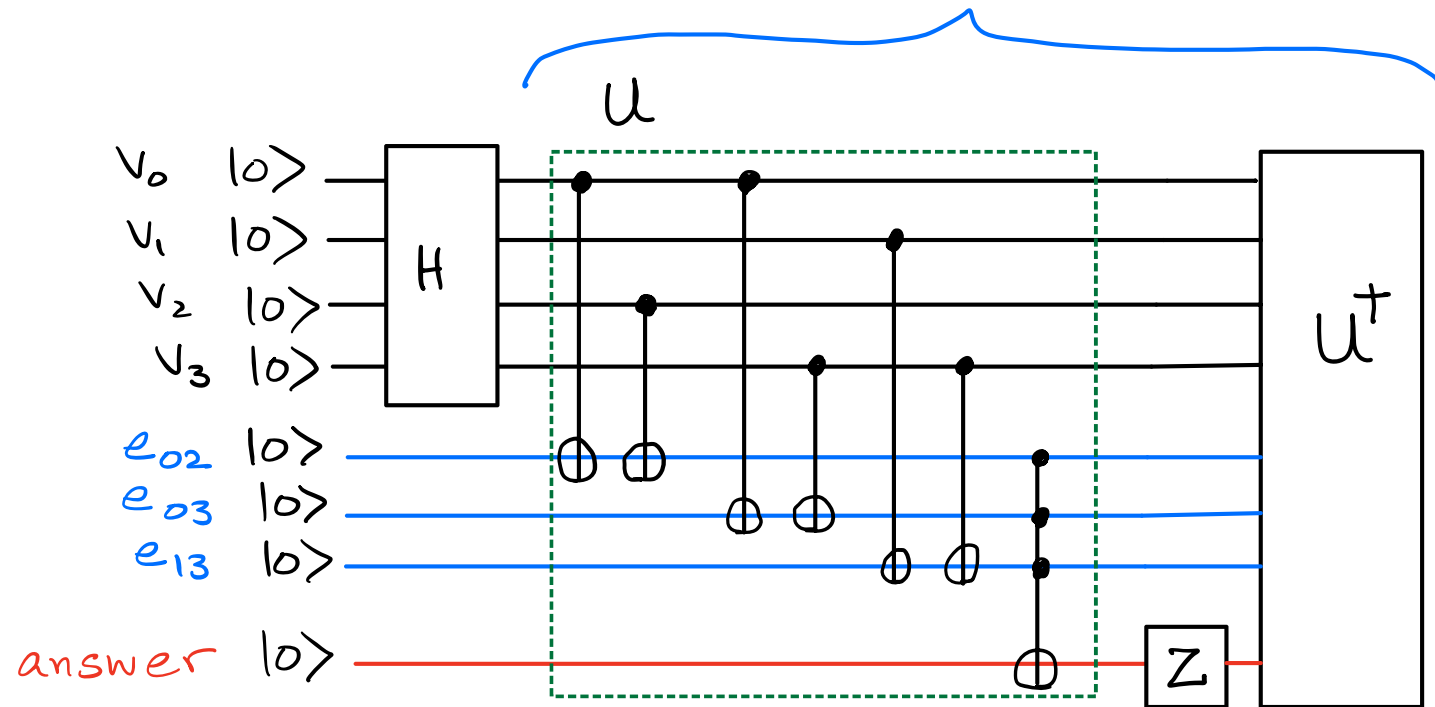


Oracle Construction





$$Z_f |x\rangle = (-1)^{f(x)} |x\rangle$$



Net result is flip of sign for qubits representing correct colorings.

This can be used as an oracle for Grover!

Checking a Graph is Bipartite

Our Quantum Algorithm: $O(\sqrt{2^n})$

Best Classical Algorithm: $O(n^2)$

General Construction

Oracle Construction:

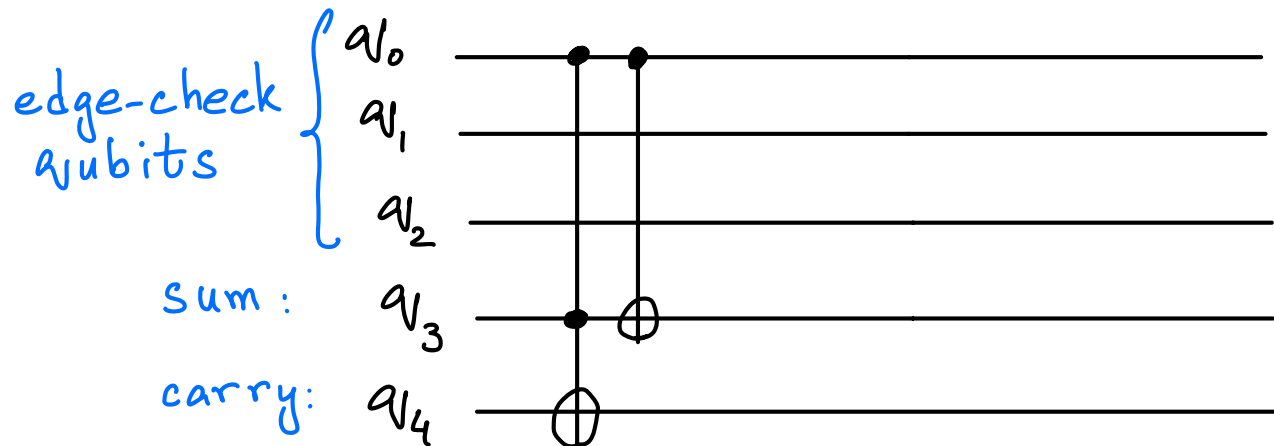
1. Edge checking for each edge
2. Sum the outputs of edge checking
3. Check whether $\text{sum} \geq k$ & store result
4. Apply Z gate on result

General Construction

Oracle Construction:

1. Edge checking for each edge
2. Sum the outputs of edge checking
3. Check whether $\text{sum} \geq k$ & store result
4. Apply Z gate on result

In-place Addition: Given 3 bits, how many are set to 1.

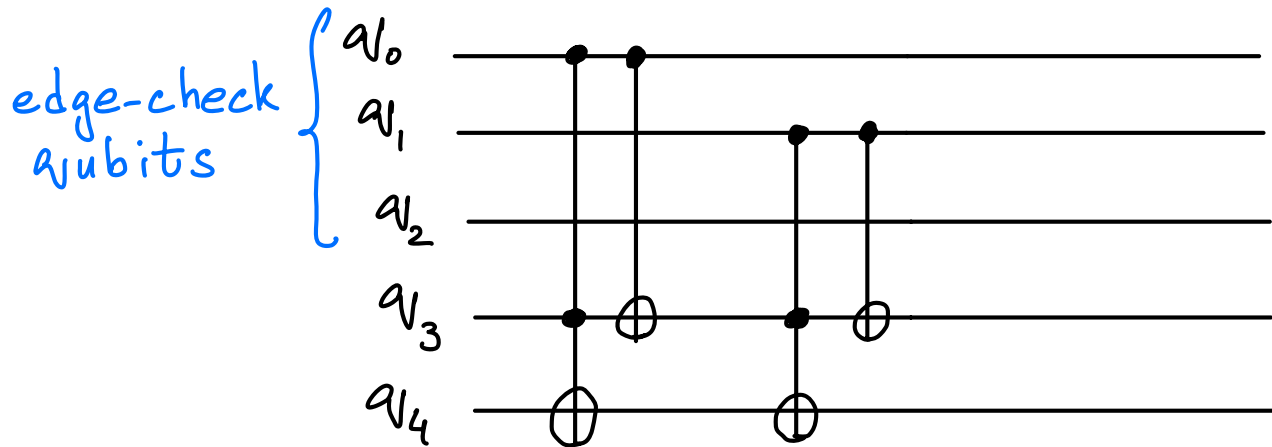


General Construction

Oracle Construction:

1. Edge checking for each edge
2. Sum the outputs of edge checking
3. Check whether $\text{sum} \geq k$ & store result
4. Apply Z gate on result

In-place Addition: Given 3 bits, how many are set to 1.

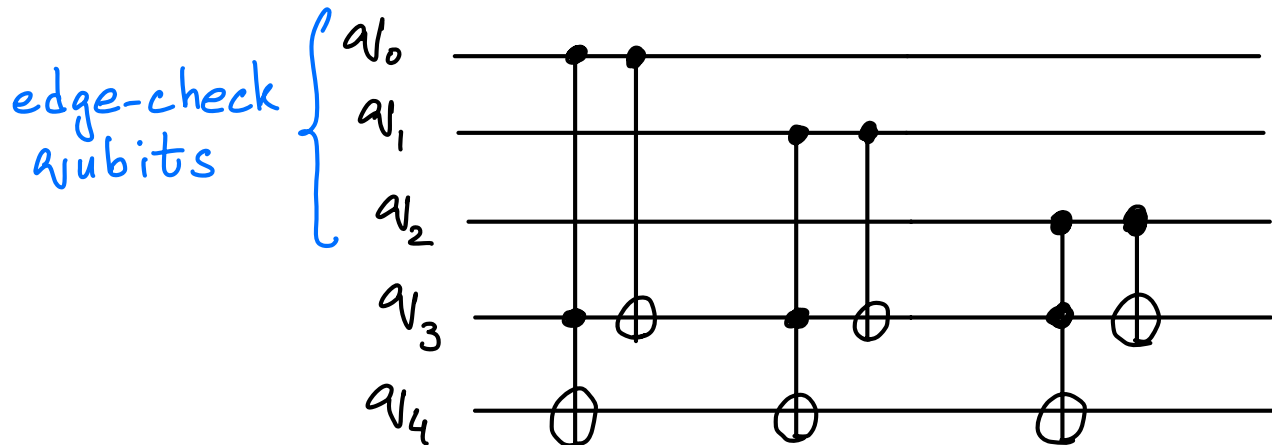


General Construction

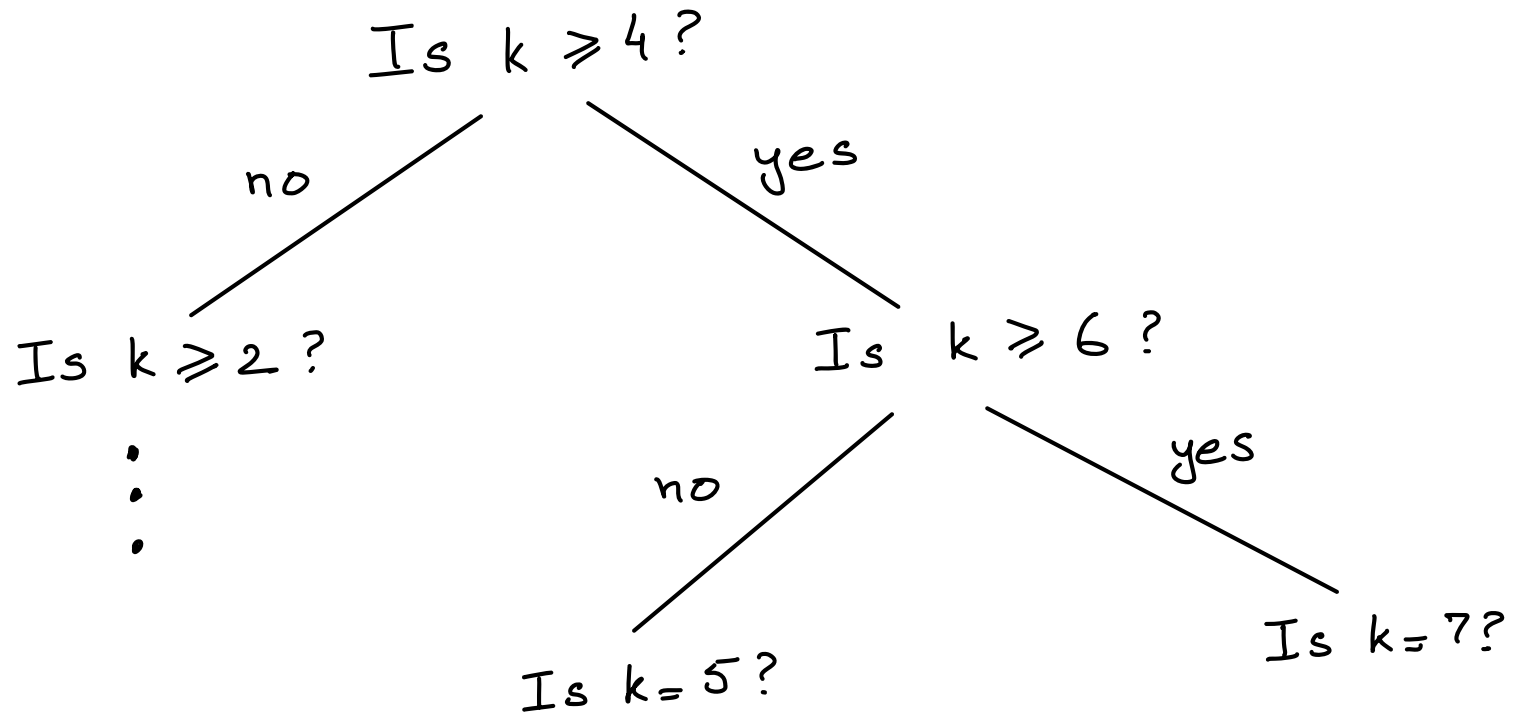
Oracle Construction:

1. Edge checking for each edge
2. Sum the outputs of edge checking
3. Check whether $\text{sum} \geq k$ & store result
4. Apply Z gate on result

In-place Addition: Given 3 bits, how many are set to 1.

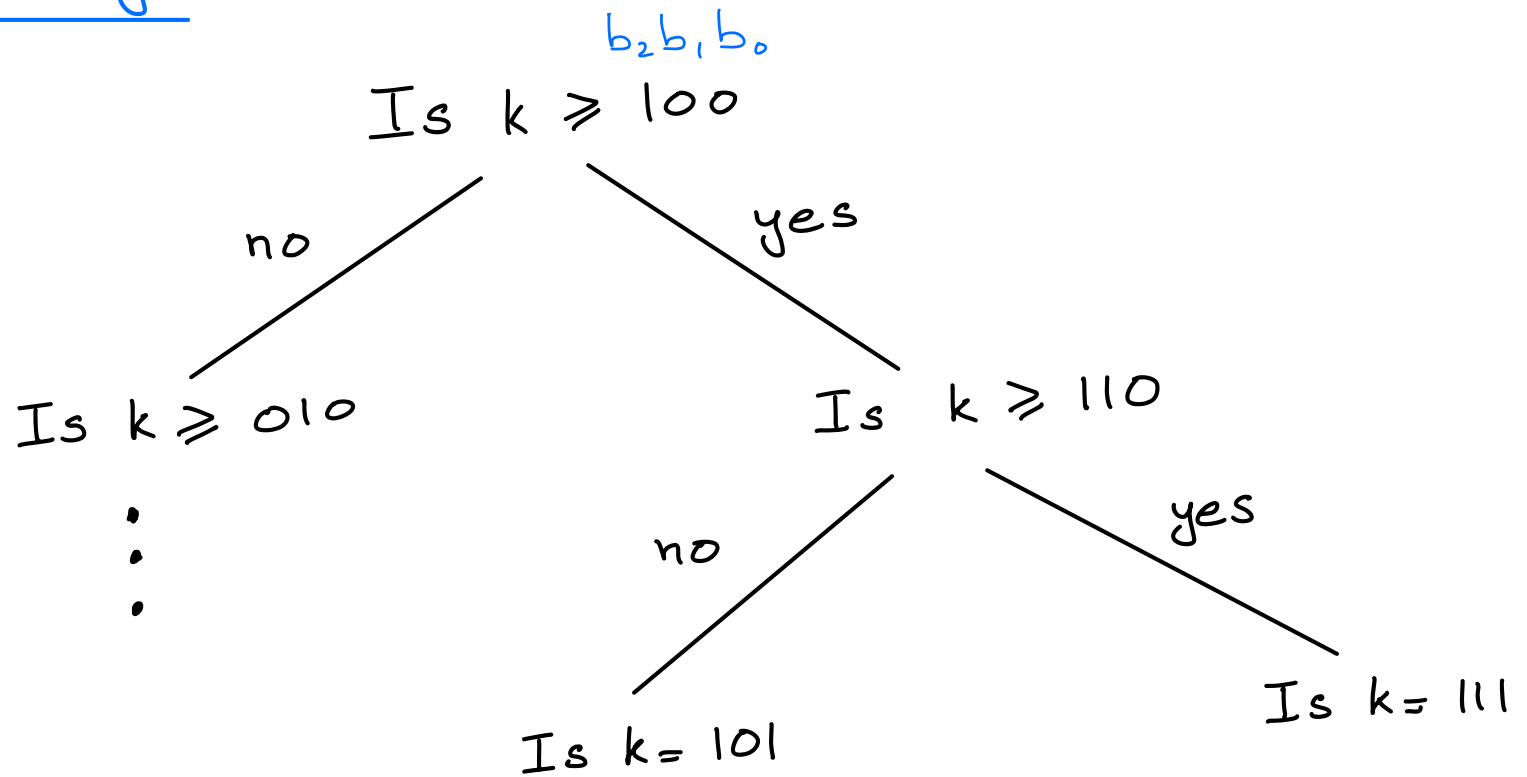


Number-Checking



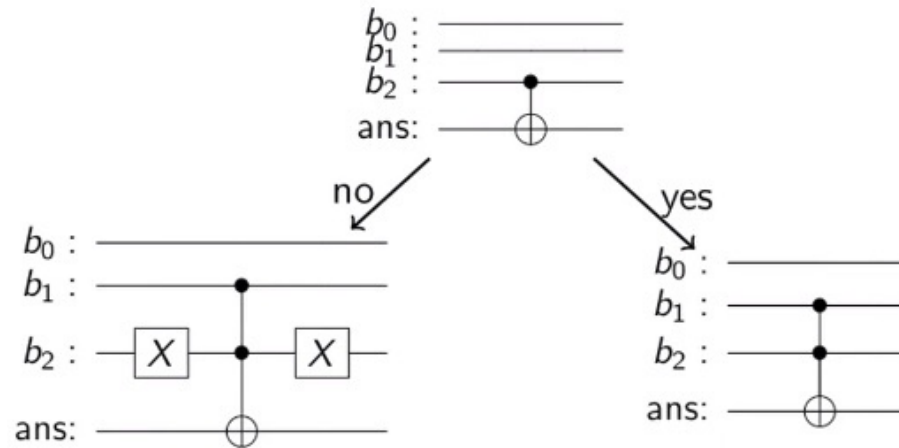
Number-Checking

(3 bits)



Number-Checking

(3 bits)



Note: Each case will require a separate Grover search!

At most $\lg(\# \text{ of edges})$

What is Your Favourite Super Power?

MANIPULATE PROBABILITY!!!