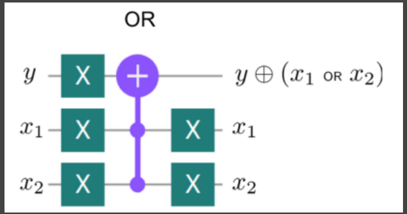
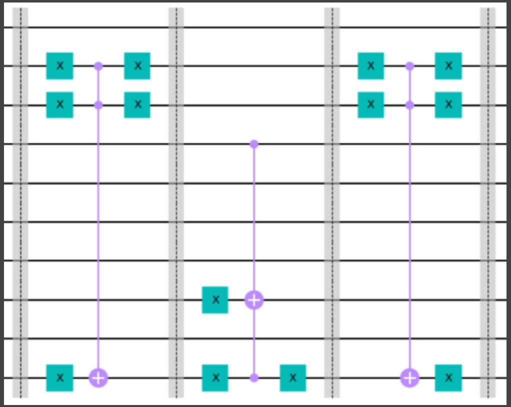
3SAT FINAL PROJECT

Michelle Ding
Letizia Fazzini

TABLE OF CONTENTS



01 QUANTUM ORACLE



- 3-SAT refresher:
    - Boolean expressions such as “(x0 or !x1 or x2) and (!x2 or x1 or !x0) and (x1 or x0 or !x2)”
    - Does a 3-SAT expression have solutions? NP-complete
    - Grover’s Algorithm can find them faster
- Turning 3-SAT expression into quantum oracle
    - Implemented with OR gates and Multi-AND gates
    - How to minimize auxiliary bits?
        - Reuse them by uncomputing parts of circuit
        - Need to reuse all auxiliary bits with Grover’s Algorithm

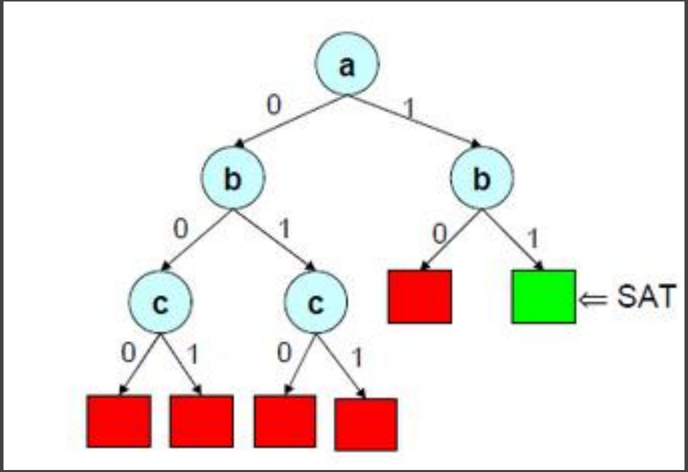
02 NO. SOLUTIONS

Is the formula is satisfiable? *

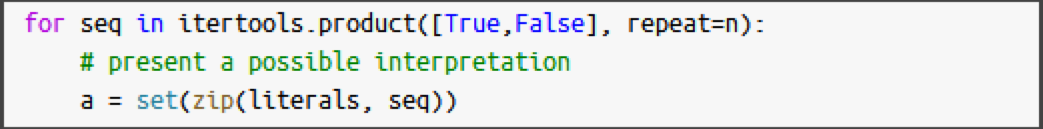
If so, we can use brute force to find the number of solutions

Many ways…we keep a set of all possible interpretations

If not, return unsatisfiable



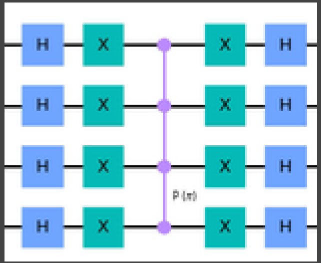
* run a basic version of the DPLL algo to find out



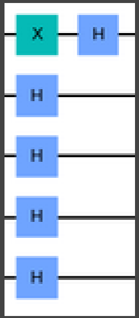
```
for seq in itertools.product([True,False], repeat=n):
    # present a possible interpretation
    a = set(zip(literals, seq))
```

03 GROVER'S ALGORITHM

initialize qc ——> Q.CIRCUIT ——>



initialize qc2 ——>

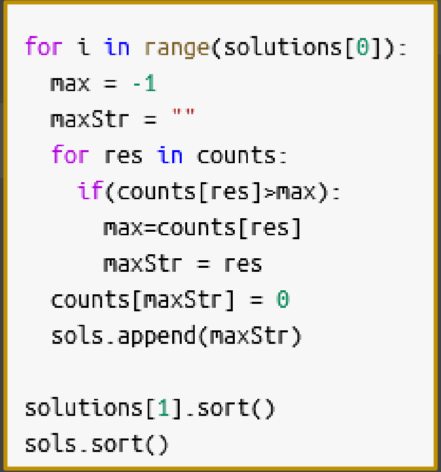


$$r = \lfloor \pi * \sqrt{\frac{2^{literals}}{classical\ sols}}\ /4 \rfloor$$

——>

qc

getting the solutions:

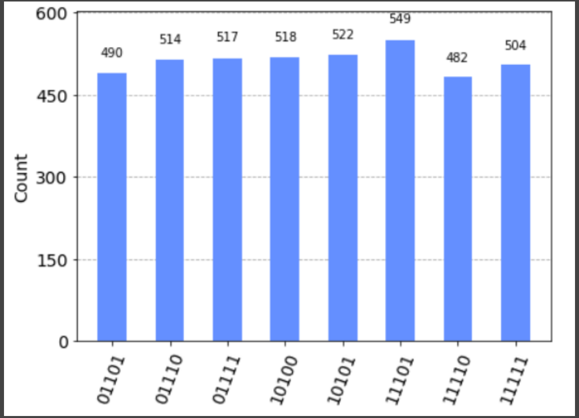


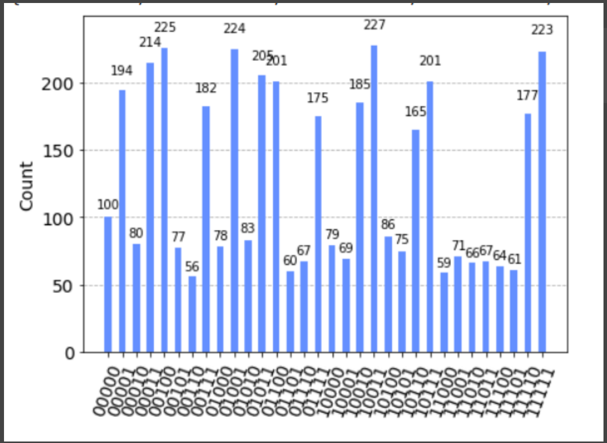
```
for i in range(solutions[0]):
  max = -1
  maxStr = ""
  for res in counts:
    if(counts[res]>max):
      max=counts[res]
      maxStr = res
  counts[maxStr] = 0
  sols.append(maxStr)

solutions[1].sort()
sols.sort()
```

04 EXPERIMENTAL RESULTS

DEMO!


Count


Count