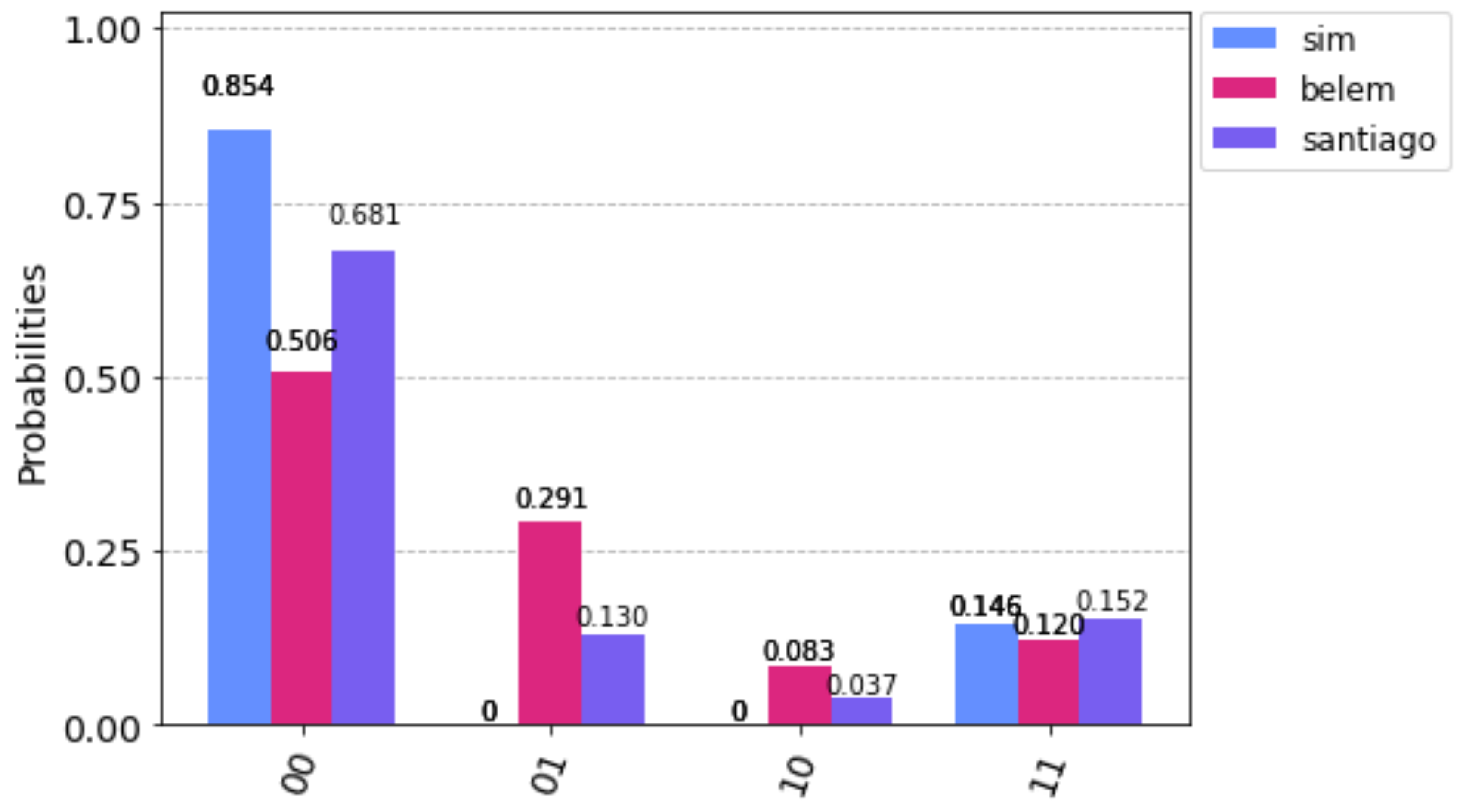
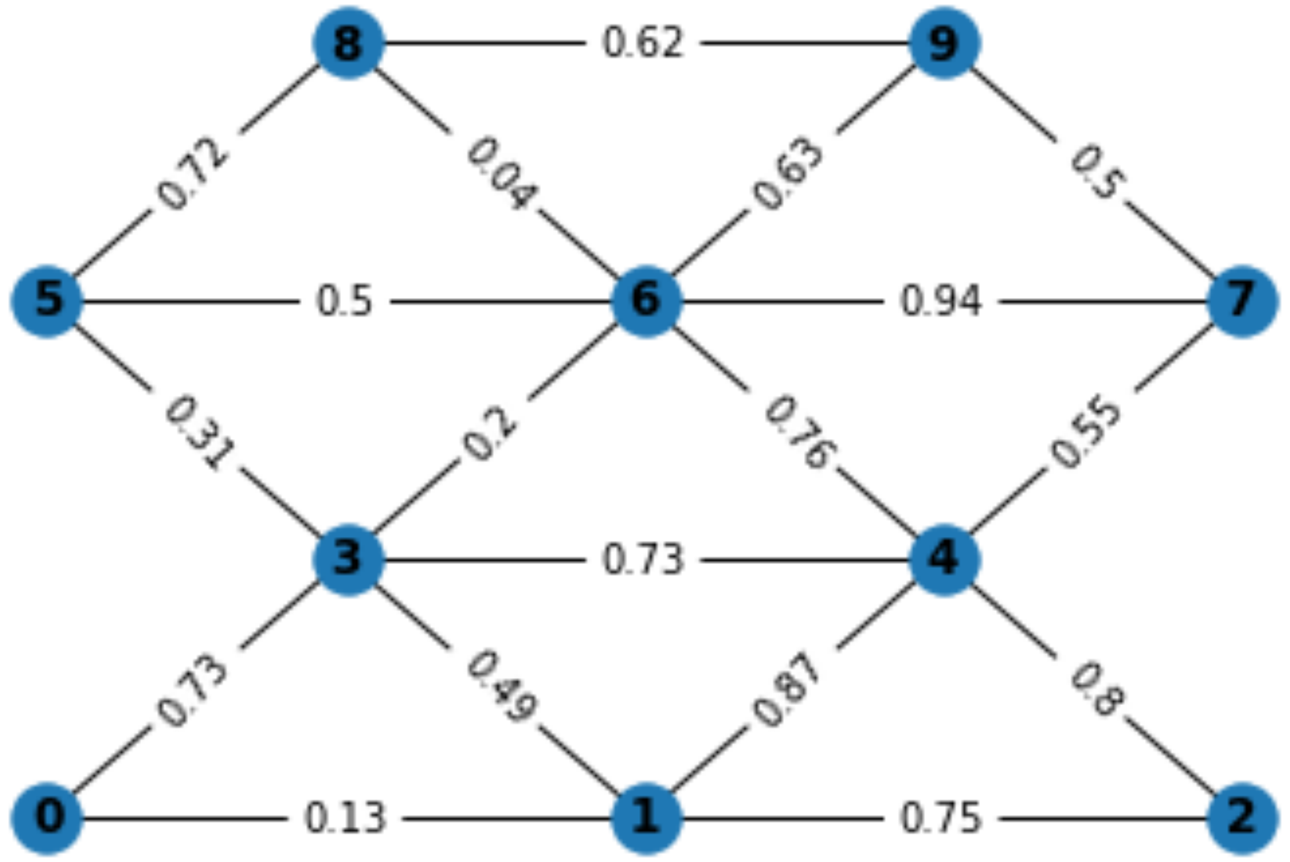
Teaching quantum computing for second year students in science and engineering

```
Job Status: job has successfully run
```

Out[30]:



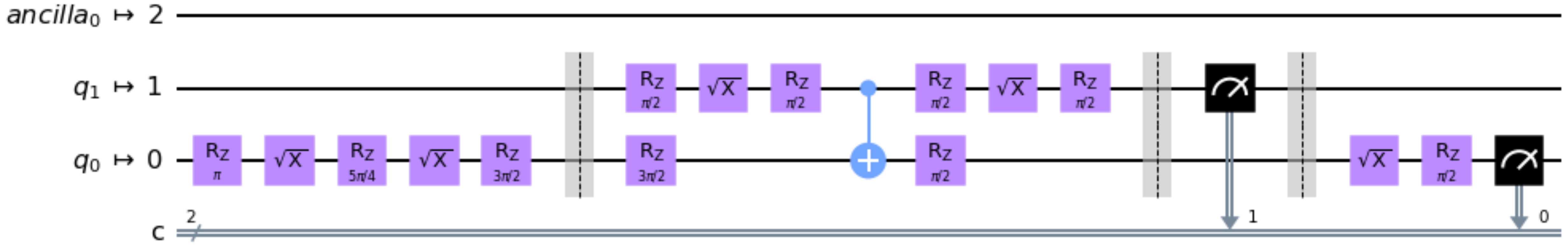


```
for sample, energy, num_occurrences, chain in response.data():
    print(sample, "Energy: ", energy, "Occurrences: ", num_occurrences)
```

```
{0: 1, 1: 1, 2: -1, 3: -1, 4: 1, 5: 1, 6: -1, 7: -1, 8: -1, 9: 1} Energy:  -5.909999999999999 Occurrences:  138
{0: -1, 1: -1, 2: 1, 3: 1, 4: -1, 5: -1, 6: 1, 7: 1, 8: 1, 9: -1} Energy:  -5.909999999999999 Occurrences:  327
{0: 1, 1: 1, 2: -1, 3: -1, 4: 1, 5: 1, 6: -1, 7: 1, 8: -1, 9: 1} Energy:  -5.69 Occurrences:  47
{0: -1, 1: -1, 2: 1, 3: 1, 4: -1, 5: -1, 6: 1, 7: -1, 8: 1, 9: -1} Energy:  -5.69 Occurrences:  185
```

In [35]:

```
# Why the devices have so much noise? Actually there is a simple reason. The native gates in belem and
# santiago are:
```



The best answer has E=-5.909999999999999 and the QPU finds this 138+327=465 out of 1000 runs (46.5%).

Rogério de Sousa

Dept. of Physics and Astronomy, University of Victoria, BC

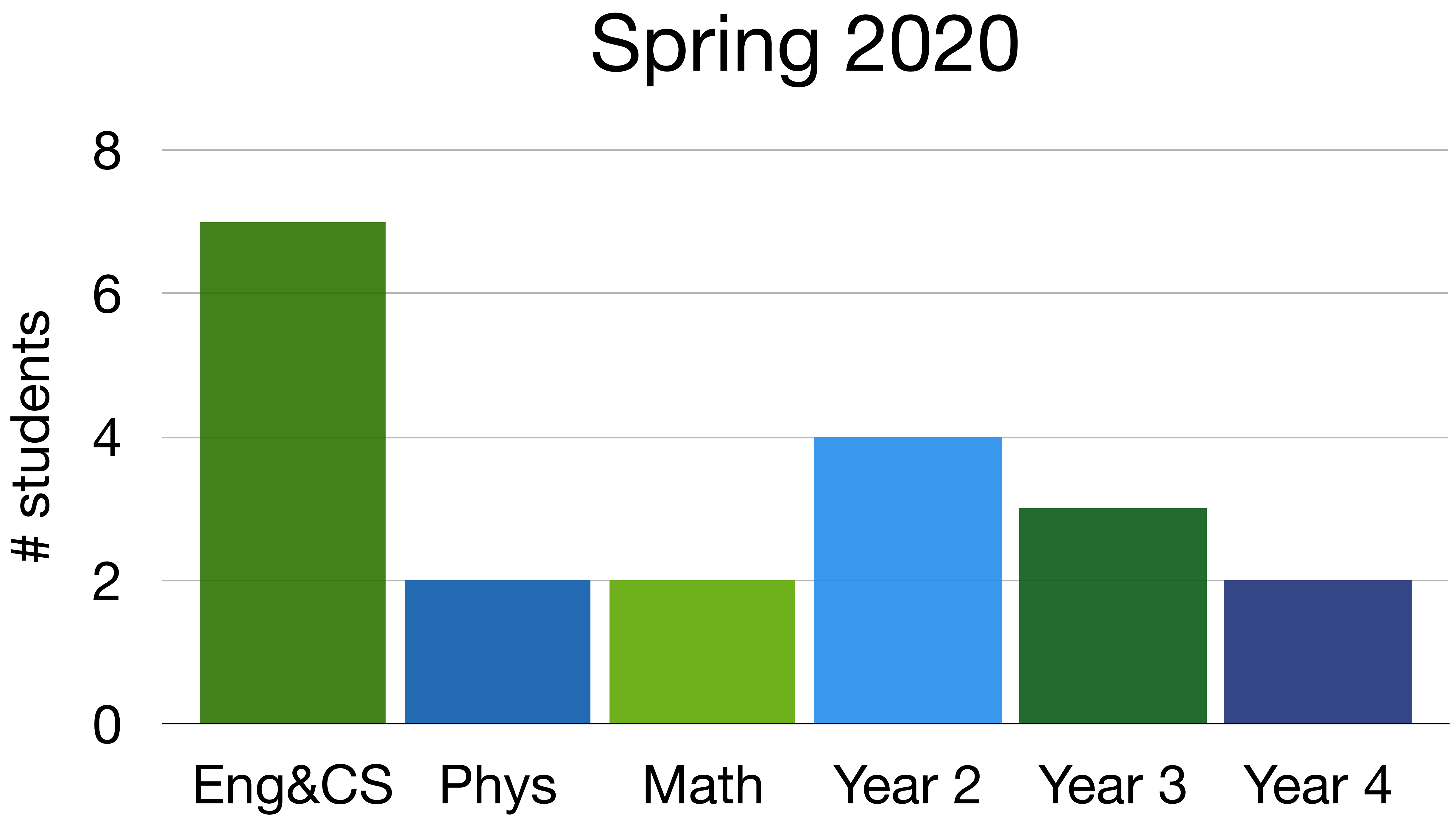
Territorial acknowledgement

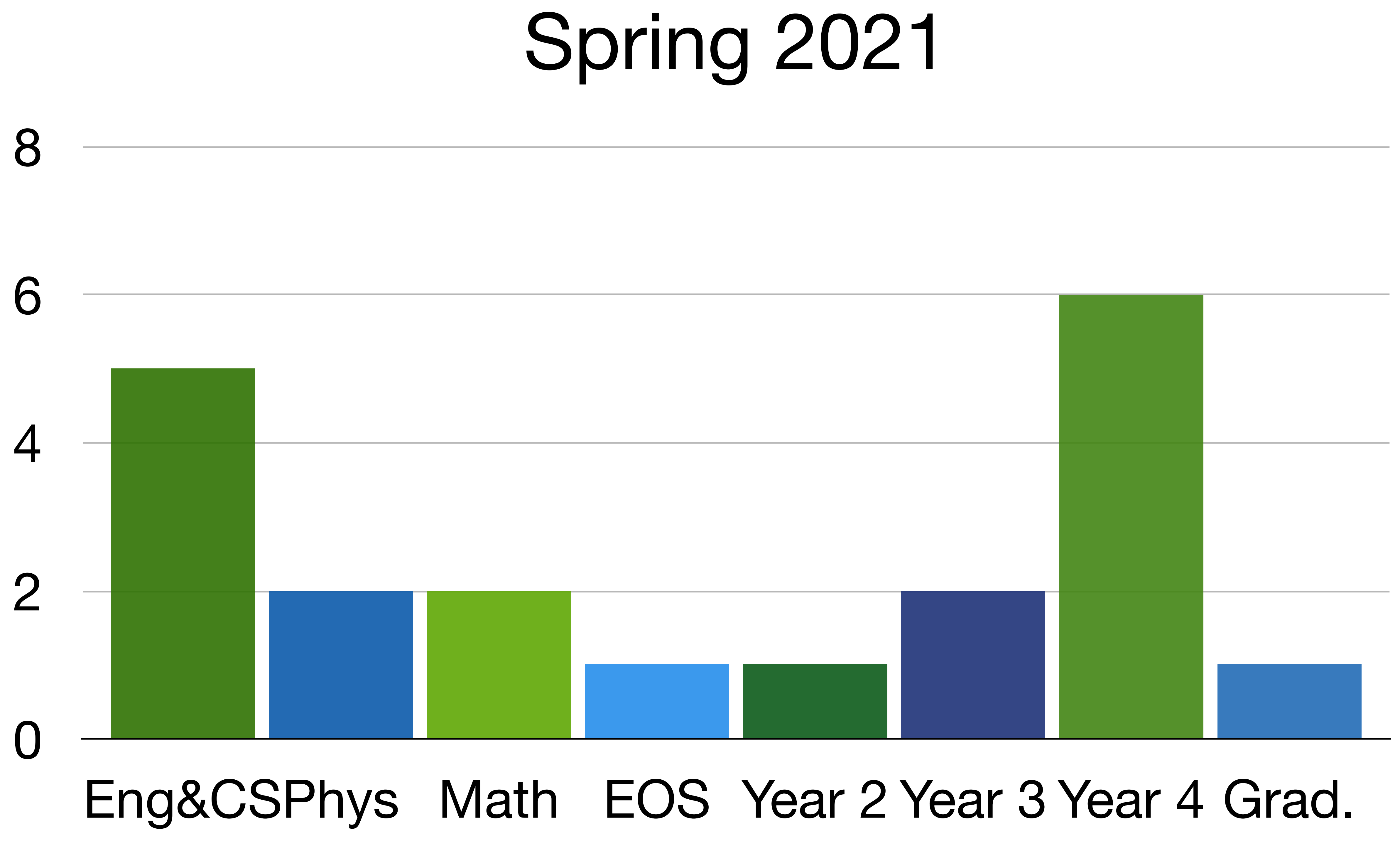
“We acknowledge and respect the lək̓ʷəŋən peoples on whose traditional territory the university stands and the Songhees, Esquimalt and W̱SÁNEĆ peoples whose historical relationships with the land continue to this day”

Pre-requisites and audience

Course taught Spring 2020, 2021

- The only pre-req was linear algebra. No background in quantum theory or Python required.
- “Quantum hype”: Class starts overbooked (20 students) and when students realize it’s serious work it decreases to 10.





Learning goals

1. Proficiency in quantum theory applied to qubits: Dirac's notation, unitary transformations, superposition, entanglement, measurement.

2. Ability to write a quantum circuit that prepares a specific few qubit state. Explain the notions of universality and circuit depth.

3. Demonstrate basic quantum algorithms and the speed up they achieve: Deutch's, Simon's, Quantum phase estimation. “Trade off” between parallelism and measurement. Debunk popular vulgarization that QC can solve 2^n inputs in one go.

4. Demonstrate simple cryptography and quantum key distribution using BB84. Teleportation and elementary error correction (bit-flip code with 3 qubits).

5. Ability to code any quantum circuit using IBM's Qiskit. Make appropriate measurements and interpret data for simulators and real devices. Explain limitations of real devices: Noise, basic notion of quantum volume.

6. Elementary demonstration of adiabatic quantum computing: Hamiltonian operator, ground and excited energy eigenstates. Application to the 3-SAT problem. Quantum annealing as heuristic-non-universal AQC.

7. Ability to minimize a general QUBO using D-Wave Leap. Impact of device topology, chaining, embedding.

How I did it

1. Introduced quantum theory (Stern-Gerlach, analogy with polarization of light).
2. Tools of the trade: Linear algebra with complex numbers, Dirac's notation, unitary operators.
3. Quantum measurement and the meaning of a quantum state.
4. Quantum circuits and simple algorithms (Deutch's, Bernstein-Vazirani, Simon's, BB84, teleportation, 3-bit flip error correction, phase estimation without QFT).
5. Experiential learning with IBM-Q.
6. Adiabatic quantum computing: A hook to introduce Hamiltonians and Schröedinger's eqn.; quantum annealing as heuristic.
7. Experiential learning with D-Wave Leap.

Textbooks and sources of inspiration

- My take: "Theoretical QC" is mature, and there are a lot of materials out there. What is not mature is "experiential QC": Write code, submit jobs to hardware and interpret results on noisy devices. Also teaching applications that seem to be useful in the real world.

- Suggested textbook to complement lectures:
  A First Introduction to Quantum Computing and Information, Zygelman (available for free as ebook from libraries with Springer Link).

- Invaluable sources of materials and inspiration:

  - *Modern Quantum Mechanics*, Sakurai (Yes, distilled for 2nd year's!)

  - *Quantum Computation and Quantum Information*, Nielsen & Chuang

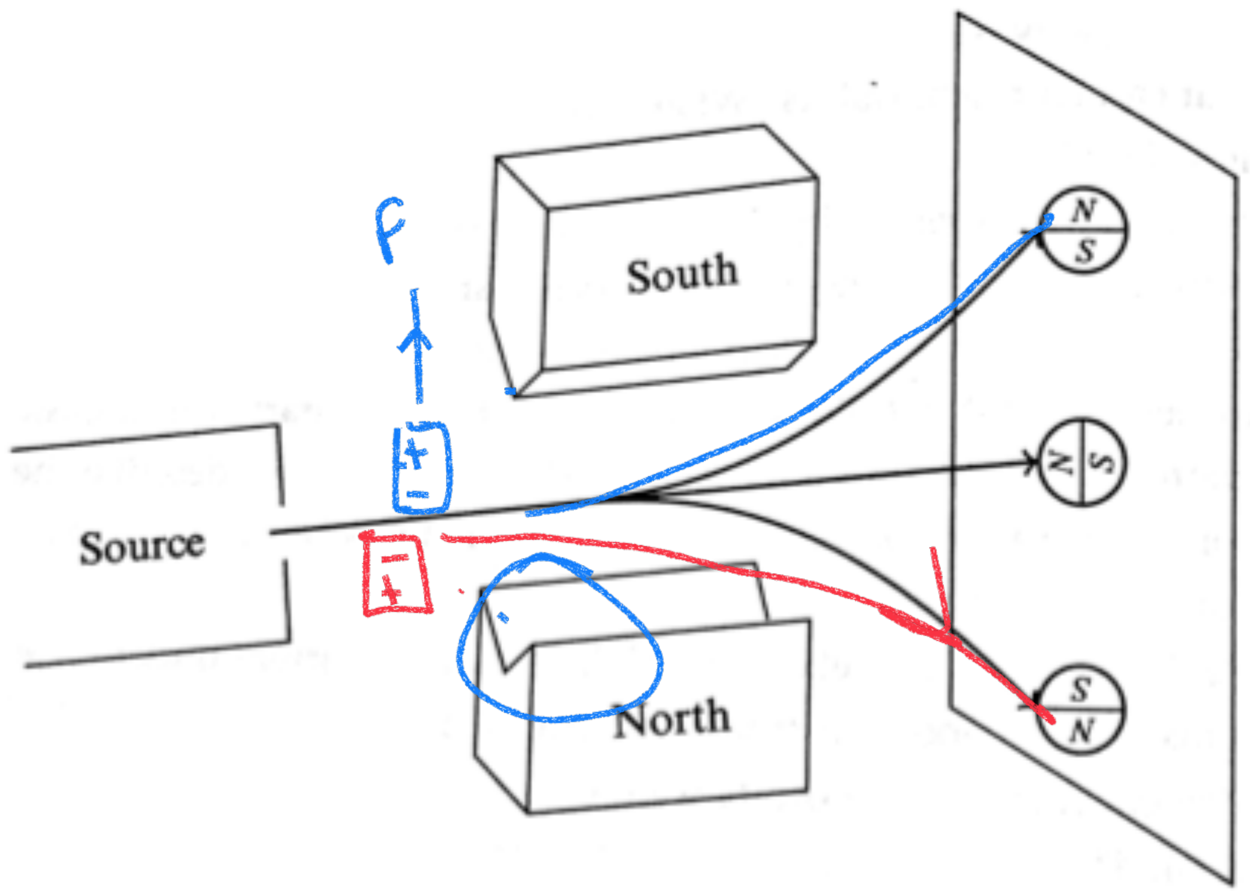
  - *Quantum Computer Science: An Introduction*, Mermin

  - Albash and Lidar, *Adiabatic QC*, Rev. Mod. Phys. **90**, 015002 (2018)

Introducing quantum theory

Sakurai's Ch. 1 distilled for 2nd years: Stern-Gerlach experiment, electron spin and analogy with polarization of light.

What is the result of this experiment?
Separate into groups of 5 and discuss.

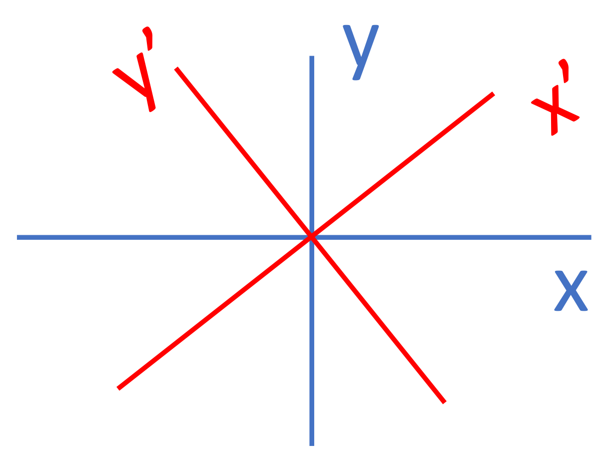


Hints:

- The Ag atoms come out of the oven with random magnetic orientation.
- The force that the shaped magnet induces on each atom is proportional to the projection of the atom's magnetic moment along the vertical.

Explanation of S-G: Analogy with polarization of light

$$S_z+ \text{ atoms} \longleftrightarrow \text{x-polarized light}$$
$$S_z- \text{ atoms} \longleftrightarrow \text{y-polarized light}$$



$$S_x+ \text{ atoms} \longleftrightarrow \text{x'-polarized light}$$
$$S_x- \text{ atoms} \longleftrightarrow \text{y'-polarized light}$$

- But, for light we know that:

$$E_0\hat{\boldsymbol{x}}'\cos(kz-\omega t) = E_0\left[\frac{1}{\sqrt{2}}\hat{\boldsymbol{x}}\cos(kz-\omega t) + \frac{1}{\sqrt{2}}\hat{\boldsymbol{y}}\cos(kz-\omega t)\right]$$

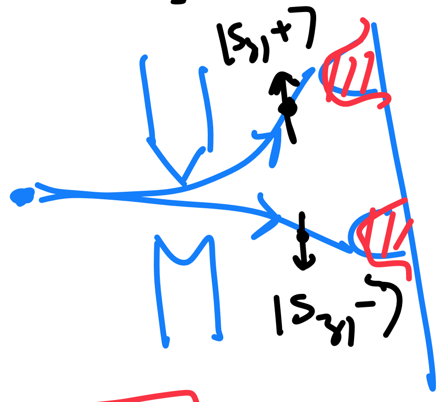
- This suggests the following *conjecture*:

$$|S_x;+\rangle = +\frac{1}{\sqrt{2}}|S_z;+\rangle + \frac{1}{\sqrt{2}}|S_z;-\rangle$$

Tools of the trade

Linear algebra with complex numbers, Dirac's notation, unitary operators

What we learned from S-G: Quantum two-level systems (e.g. spin of the electron) exist in nature



Spin of electron is a "natural" *qubit*

$$\begin{cases} |S_z, +\rangle \equiv |0\rangle \\ |S_z, -\rangle \equiv |1\rangle \end{cases}$$

- Qubit = Quantum two-level system. S-G shows that they can be in any state described by

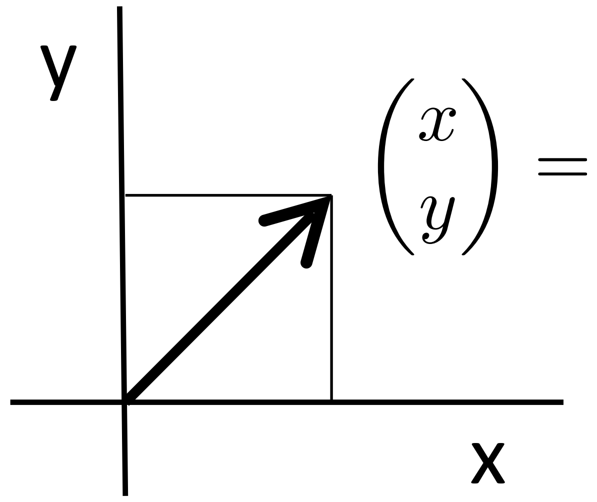
SUPERPOSITION OF $|0\rangle$ AND $|1\rangle$ COEFFICIENTS ARE COMPLEX NUMBERS

$$\boxed{|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle} \qquad \boxed{\alpha, \beta \in \mathbb{C}}$$

- We say that "spin is quantized". If we "read-out" the state of the spin qubit we will get "0" with probability $|\alpha|^2$ and "1" with probability $|\beta|^2$.

Linear algebra with Dirac's notation

- A simple vector space: $\mathbb{R}^2$



$$\begin{pmatrix} x \\ y \end{pmatrix} = x \underbrace{\begin{pmatrix} 1 \\ 0 \end{pmatrix}}_{=\hat{i}} + y \underbrace{\begin{pmatrix} 0 \\ 1 \end{pmatrix}}_{=\hat{j}} = x\hat{i} + y\hat{j}$$

- The vector space for one qubit , $\mathbb{C}^2$. This is where the "ket" lives:

$$|\Psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha \underbrace{\begin{pmatrix} 1 \\ 0 \end{pmatrix}}_{\equiv|0\rangle} + \beta \underbrace{\begin{pmatrix} 0 \\ 1 \end{pmatrix}}_{\equiv|1\rangle} = \alpha|0\rangle + \beta|1\rangle$$

- The qubit's "co-vector" or dual space, where the "bra" lives:

$\dagger$ "DAGGER"

$$\langle\Psi| = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}^{\dagger} = (\alpha^* \;\; \beta^*) = \alpha^* \underbrace{(1 \;\; 0)}_{\equiv\langle 0|} + \beta^* \underbrace{(0 \;\; 1)}_{\equiv\langle 1|} = \alpha^*\langle 0| + \beta^*\langle 1|$$

Quantum measurement, “collapse" of state, and meaning

Quantum measurement in the computational basis

POSTULATE:

- Each time a qubit is measured, the outcome is either 0 or 1, and the state of the qubit *collapses* to $|0\rangle$ or $|1\rangle$.

$$\boxed{|\Psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle} \boxed{-\!\!\boxed{\mathbf{M}}\!\!\rightarrow} |x\rangle \begin{cases} x = 0 \text{ with probability } \boxed{|\langle 0|\Psi\rangle|^2} = |\alpha_0|^2 \\ x = 1 \text{ with probability } |\langle 1|\Psi\rangle|^2 = |\alpha_1|^2 \end{cases}$$

$$\Rightarrow \boxed{|\alpha_0|^2 + |\alpha_1|^2 = 1.}$$

- For n qubits:

$$\boxed{|\Psi\rangle = \sum_{j=0}^{2^n-1} \alpha_j|j\rangle} -\!\!\boxed{\mathbf{M}}\!\!\rightarrow |j\rangle \text{ with prob. } |\underbrace{\langle j|\Psi\rangle}|^2 = |\alpha_j|^2$$

- This implies normalization:

$$\sum_j \underbrace{|\langle j|\Psi\rangle|^2}_{=\langle j|\Psi\rangle\langle\Psi|j\rangle} = 1 = \sum_j \langle\Psi|j\rangle\langle j|\Psi\rangle = \langle\Psi|\underbrace{\left(\sum_j |j\rangle\langle j|\right)}_{=I}|\Psi\rangle = \langle\Psi|\Psi\rangle \Rightarrow \boxed{\langle\Psi|\Psi\rangle = 1}$$

Another example

- Consider $\begin{cases} |+i\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle) \\ |-i\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle) \end{cases}$

$$\langle +|\pm i\rangle = \frac{1}{\sqrt{2}}(\langle 0| + \langle 1|)\frac{1}{\sqrt{2}}\begin{pmatrix}|0\rangle \\ \pm i|1\rangle\end{pmatrix} = \frac{1}{2}(1 \pm i)$$

- Measure in the computational basis: $\rightarrow \begin{cases} p(0) = |\langle 0|\pm i\rangle|^2 = \left|\frac{1}{\sqrt{2}}\right|^2 = \frac{1}{2} \\ p(1) = |\langle 1|\pm i\rangle|^2 = \left|\pm\frac{1}{\sqrt{2}}\right|^2 = \frac{1}{2} \end{cases}$

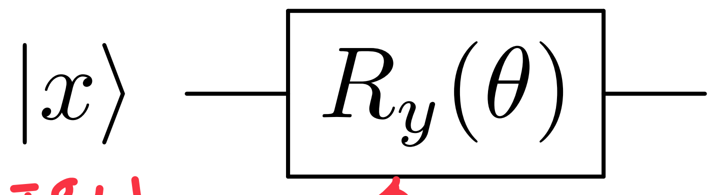
- Measure in the $\{|+\rangle, |-\rangle\}$ basis: $\longrightarrow \begin{cases} p(+) = |\langle +|\pm i\rangle|^2 = \left|\frac{1\pm i}{2}\right|^2 = \frac{1}{4}(1+i)(1-i) = \frac{1}{4}(1+\cancel{i}-\cancel{i}+1) = \frac{1}{2} \\ p(-) = |\langle -|\pm i\rangle|^2 = 1 - p(+) = \frac{1}{2} \end{cases}$

- These states are indistinguishable in both basis! That’s strange since $|\pm i\rangle$ look so similar to $|\pm\rangle$. The difference is the relative phase between $|0\rangle$ and $|1\rangle$. It has a large effect on the state!

- Meaning of quantum state: Mermin’s IQ test analogy.

Quantum circuits

Circuit for keeping a bit secret (with "one-try eavesdrop")



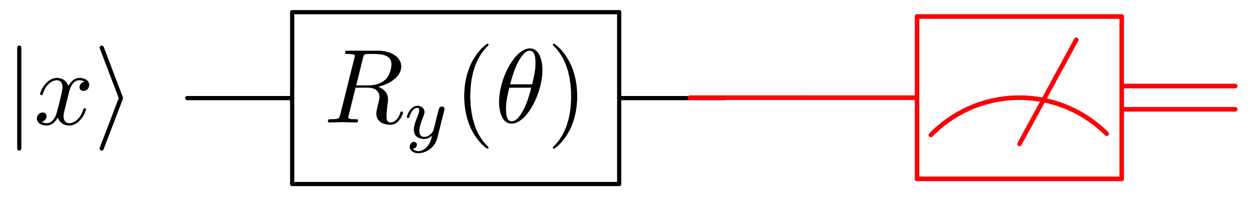
$\cos(\frac{\theta}{2})|x\rangle + (-1)^x \sin(\frac{\theta}{2})|\bar{x}\rangle$

$x = 0, 1$

"key" $\theta \in [0, 2\pi)$

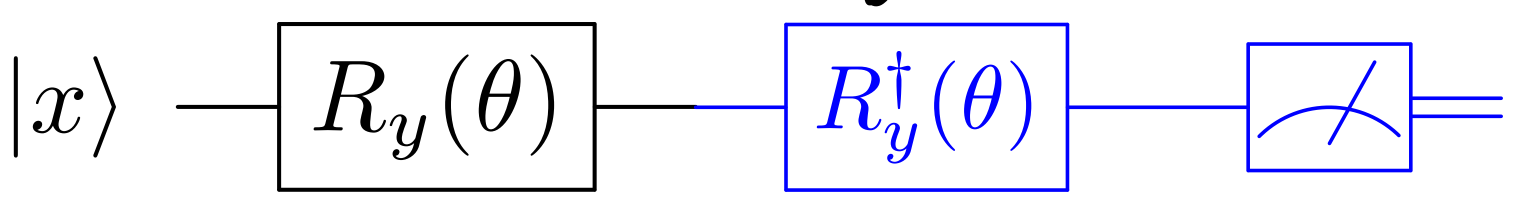
$\bar{x} = x \oplus 1$
$= NOT(x)$

- Eavesdropper:



$\begin{cases} p(x) = \cos^2(\frac{\theta}{2}) \\ p(\bar{x}) = \sin^2(\frac{\theta}{2}) \end{cases}$

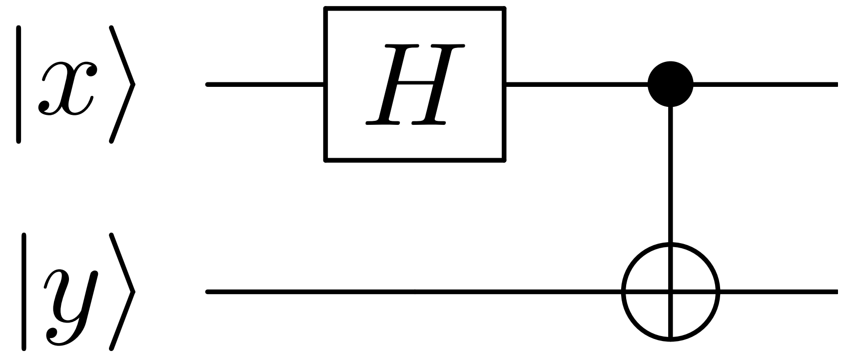
- Unlock with the "key" θ:



$\begin{cases} p(x) = 1 \\ p(\bar{x}) = 0 \end{cases}$

Time to work...

- Fill up the table



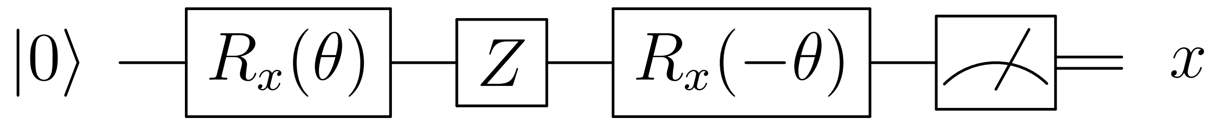
x	y	$\vert\text{output}\rangle$
0	0	$\xrightarrow{H} \vert +0\rangle = \frac{1}{\sqrt{2}}(\vert 00\rangle + \vert 10\rangle) \xrightarrow{CNOT} \frac{1}{\sqrt{2}}(\vert 00\rangle + \vert 11\rangle) = \vert\beta_{00}\rangle$
0	1	$\xrightarrow{H} \vert +1\rangle = \frac{1}{\sqrt{2}}(\vert 01\rangle + \vert 11\rangle) \xrightarrow{CNOT} \frac{1}{\sqrt{2}}(\vert 01\rangle + \vert 10\rangle) = \vert\beta_{01}\rangle$
1	0	$\xrightarrow{H} \vert -0\rangle = \frac{1}{\sqrt{2}}(\vert 00\rangle - \vert 10\rangle) \xrightarrow{CNOT} \frac{1}{\sqrt{2}}(\vert 00\rangle - \vert 11\rangle) = \vert\beta_{10}\rangle$
1	1	$\xrightarrow{H} \vert -1\rangle = \frac{1}{\sqrt{2}}(\vert 01\rangle - \vert 11\rangle) \xrightarrow{CNOT} \frac{1}{\sqrt{2}}(\vert 01\rangle - \vert 10\rangle) = \vert\beta_{11}\rangle$

$\{|\beta_{xy}\rangle\}$ IS CALLED "BELL BASIS".

(AFTER JOHN S. BELL)

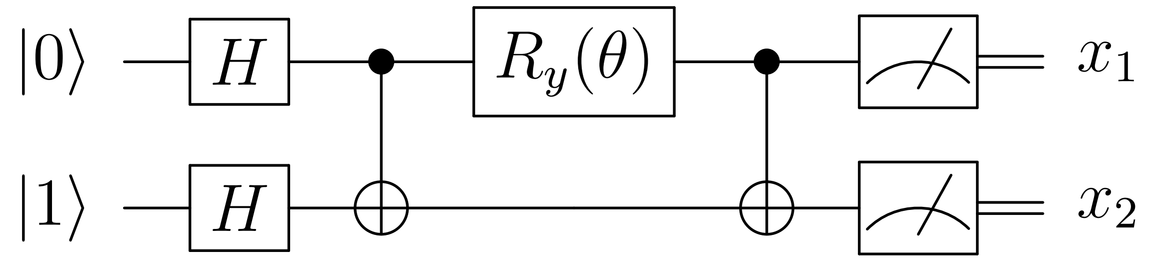
Midterm exam

4. **(/6) Probability outcomes for a one-qubit circuit.**– What are the probability outcomes $p(x)$ for the circuit



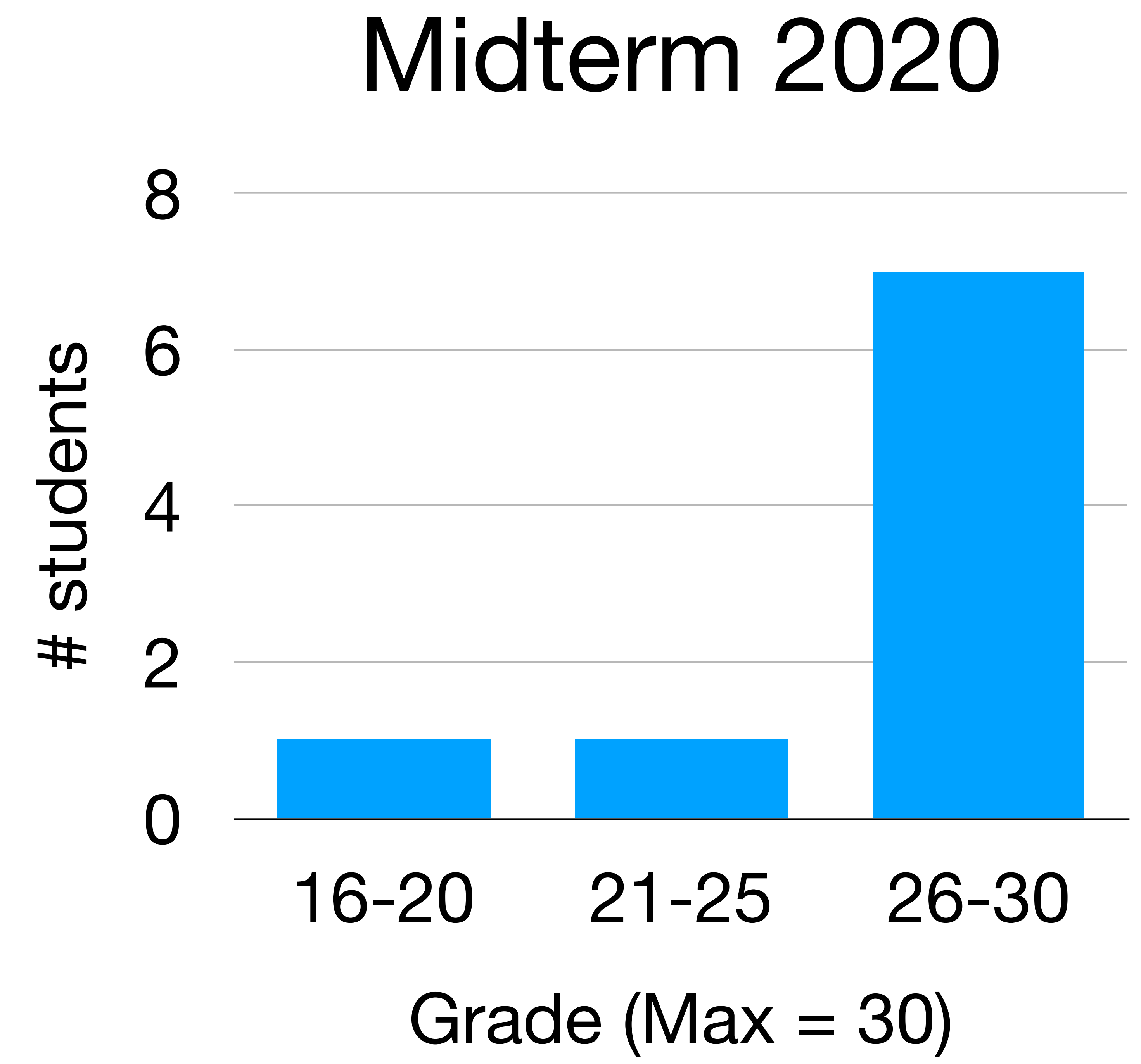
(4)

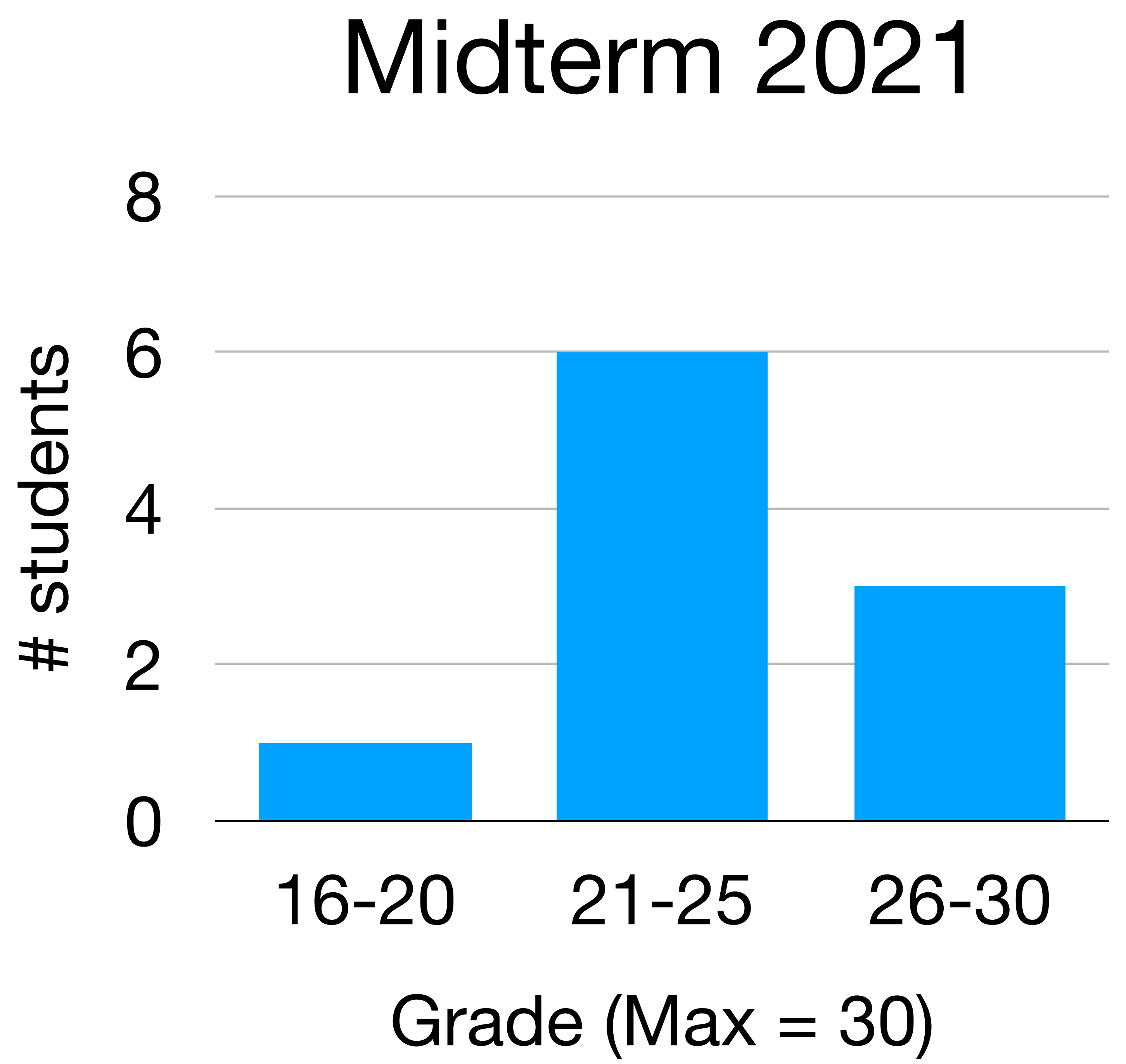
5. **(/6) Probability outcomes for a two-qubit circuit.**– What are the probability outcomes $p(x_1, x_2)$ for the circuit:



(5)

Hint: The easiest way is to compute the state after each gate is applied, using Dirac's notation.

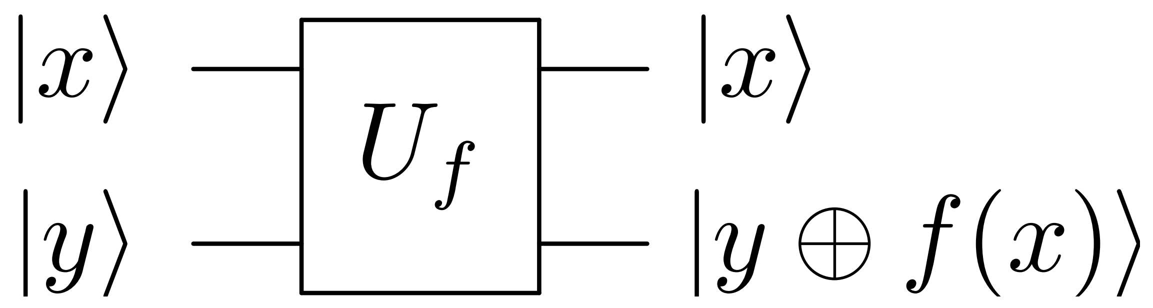




Simple algorithms

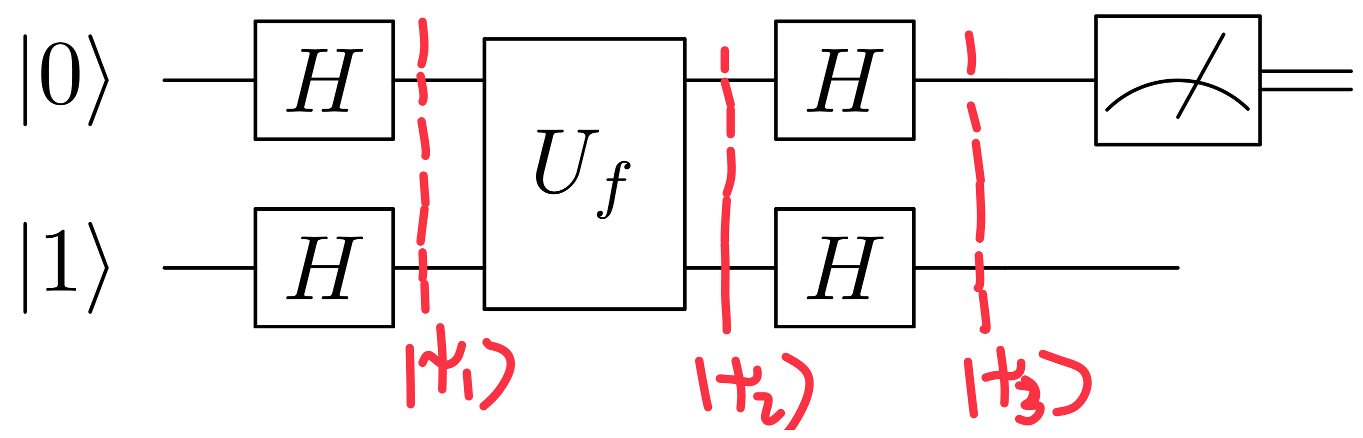
- Concept of universality for both classical and quantum computers. Irreversible vs. reversible gates.

Programming a QC to evaluate Boolean f(x)



- Deutch's algorithm: Quantum parallelism with "trade-off" due to measurement.

Deutch's algorithm for f:{0,1}→{0,1}



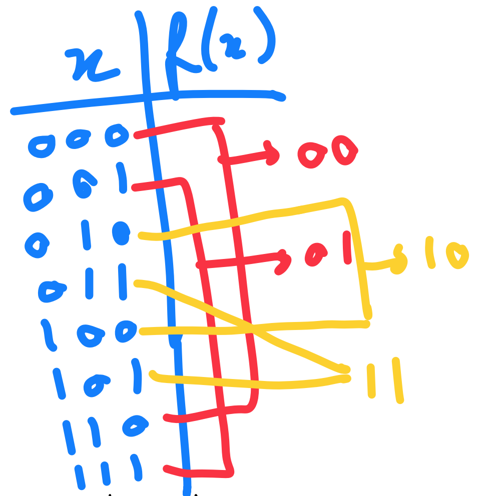
$$\begin{cases} x=0 \Rightarrow f \text{ IS CONST.} \\ x=1 \Rightarrow f \text{ IS BALANCED.} \end{cases}$$

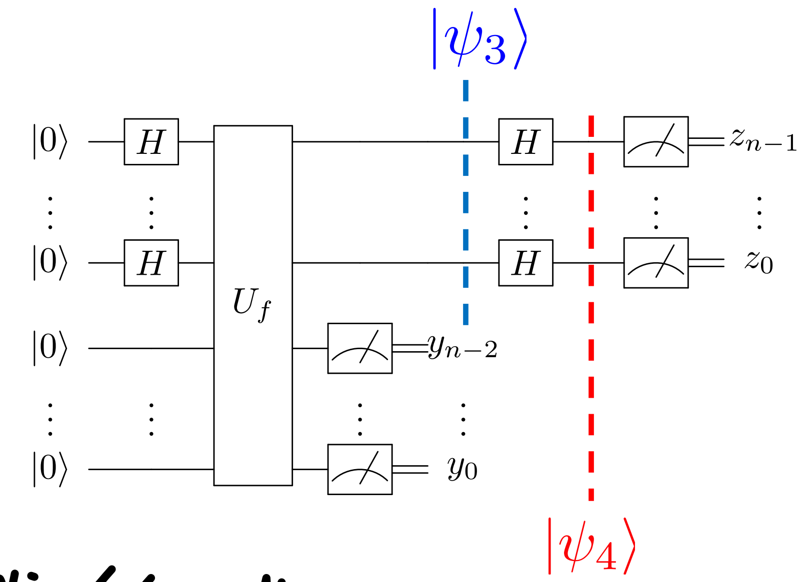
Other algorithms: Simon's, Q. key dist. with BB84

Let's work a particular case

- For n=3, choose your favourite "a" and "f(x)". Work through Simon's.

1) Choose "a" $a = 110$
2) Write down the table for f(x)
3) Assume a particular "y" outcome, Calculate $|\psi_3\rangle$ and $|\psi_4\rangle$
4) Assume a particular "z" outcome, reduce "a"
5) Repeat until you find "a"





1ST RUN: GET $y=11$

$$|\psi_3\rangle = \frac{1}{\sqrt{2}}\left(\underbrace{|011\rangle}_{x} + \underbrace{|101\rangle}_{x\oplus a}\right)$$

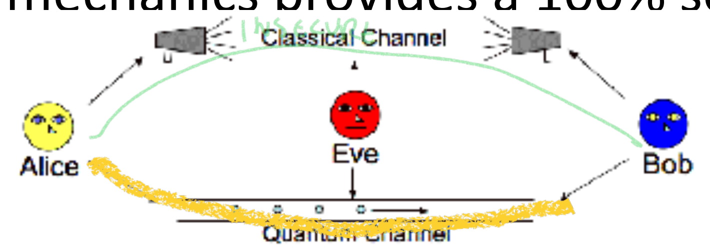
$$|\psi_4\rangle = \frac{1}{\sqrt{2^2}} \sum_{a\cdot z=0} (-1)^{x\cdot z}|z\rangle_n$$

$$= \frac{1}{2}\left(|000\rangle + (-1)^1|001\rangle + (-1)^1|110\rangle + (-1)^0|111\rangle\right)$$

$p=3/4$ TO GET USEFUL z!

Quantum key distribution: BB84 protocol (Bennet, Brassard 1984)

- Problem: How can Alice and Bob share a key remotely without allowing Eve to learn the key? Quantum mechanics provides a 100% secure method for that!



- BB84: Alice sends Bob a sequence of qubits randomly chosen to be in one of four states:

$$\underbrace{|0\rangle, |1\rangle}_{\text{TYPE 1}}, \underbrace{H|0\rangle, H|1\rangle}_{\text{TYPE H}}$$

- As each qubit arrives Bob randomly decides whether to measure in the type-1 basis (with an x-polarizer) or in the type-H basis (with a 45^0 polarizer).

BB84

- After Bob measured all the qubits (and recorded his sequence of 0s and 1s), Alice tells him over an insecure channel which qubits were sent type-1 and which type-H. However, *she does not reveal* which state she prepared for each qubit: Whether it was |0>, |1>, or H|0>, H|1>.
- On average, half of Bob's choice of basis will coincide with Alice's choice of basis. For those qubits (the "coincidence qubits"), Bob will learn the actual random bit 0 or 1 that Alice chose to send. Finally, Bob tells Alice over an insecure channel which qubits coincided (without revealing their state!). Now they both share a list of ~ n/2 random bits that they can use as a one-time pad!

		1	2	3	4	5	6	7	8	9 . . .
Alice:	Type:	1	H	H	H	1	1	H	1	H . . .
	State:	0	1	0	1	1	0	1	0	0 . . .
Bob:	Measurement type:	H	H	H	1	1	H	1	1	1 . . .
	Outcome:	1	1	0	0	1	1	1	0	0 . . .

ONE-TIME PAD:
$S = 1010 \ldots$

Experiential learning with IBM-Q

Introduction to Quantum Computing

Assignment 5 - Due March 26

Quantum algorithms with IBM-Q

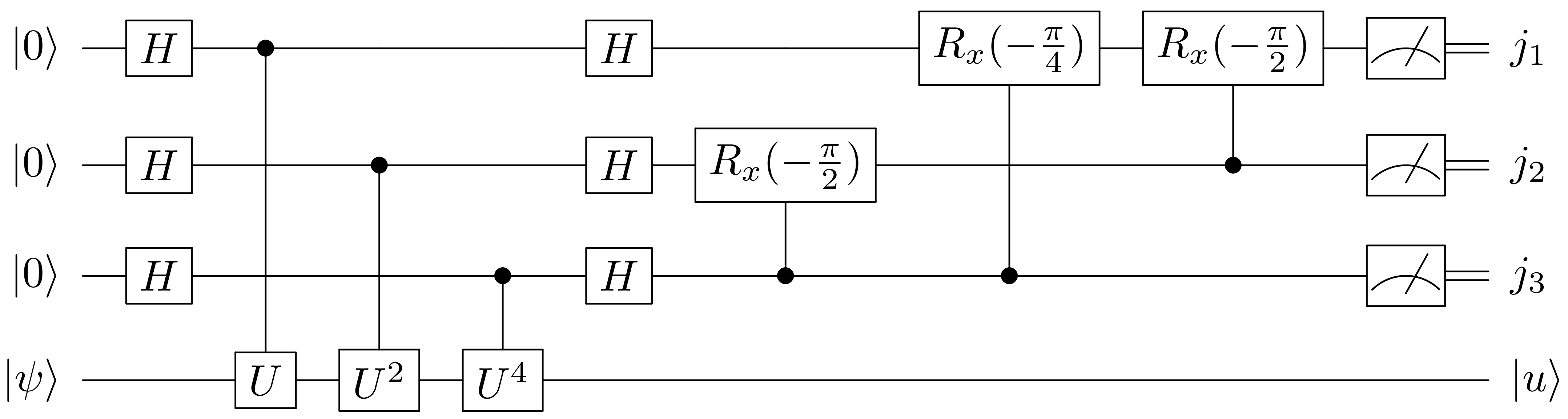
5. **Three-bit phase estimation**.

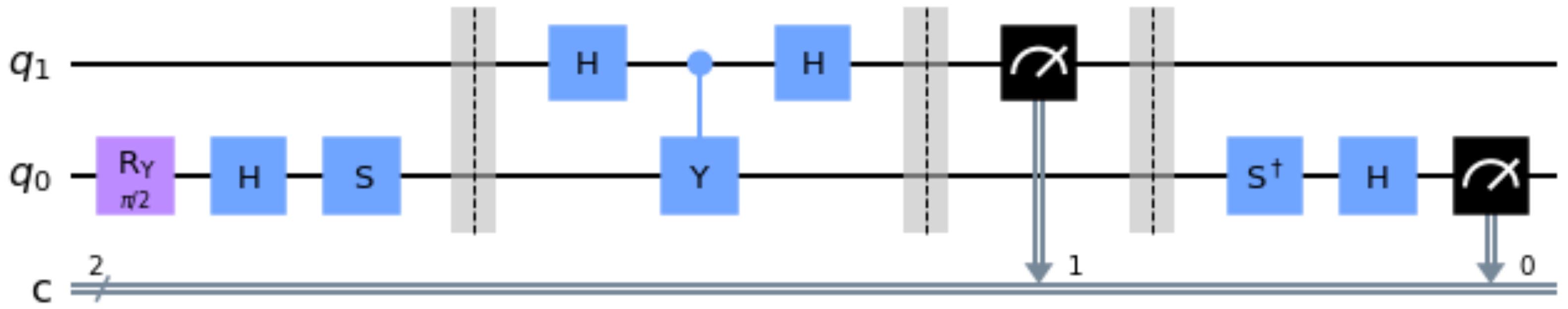
   (a) Implement the circuit below for $U = R_y(\theta)$ with $\frac{\theta}{4\pi}$ exactly represented by three base-2 decimals, $\frac{\theta}{4\pi} = 0.j_1j_2j_3$ (your choice of $j_1j_2j_3$). Note that $\frac{\theta}{4\pi}$ is the associated phase ϕ_u for $|u\rangle = |-i\rangle$, since

$$R_y(\theta)\,|-i\rangle = \mathrm{e}^{+i\frac{\theta}{2}}\,|-i\rangle = \mathrm{e}^{+2\pi i\frac{\theta}{4\pi}}\,|-i\rangle\,. \tag{5}$$

   Input $|\psi\rangle = |-i\rangle$, and run the circuit in the simulator and a real device to compare.

   (b) Run the circuit again for $\frac{\theta}{4\pi} = 0.j_1j_2j_3 - \epsilon$, with ϵ small. If you run the circuit many times does it return the $0.j_1j_2j_3$ closest to the correct answer $\frac{\theta}{4\pi}$? How does the real device compare to the simulator?

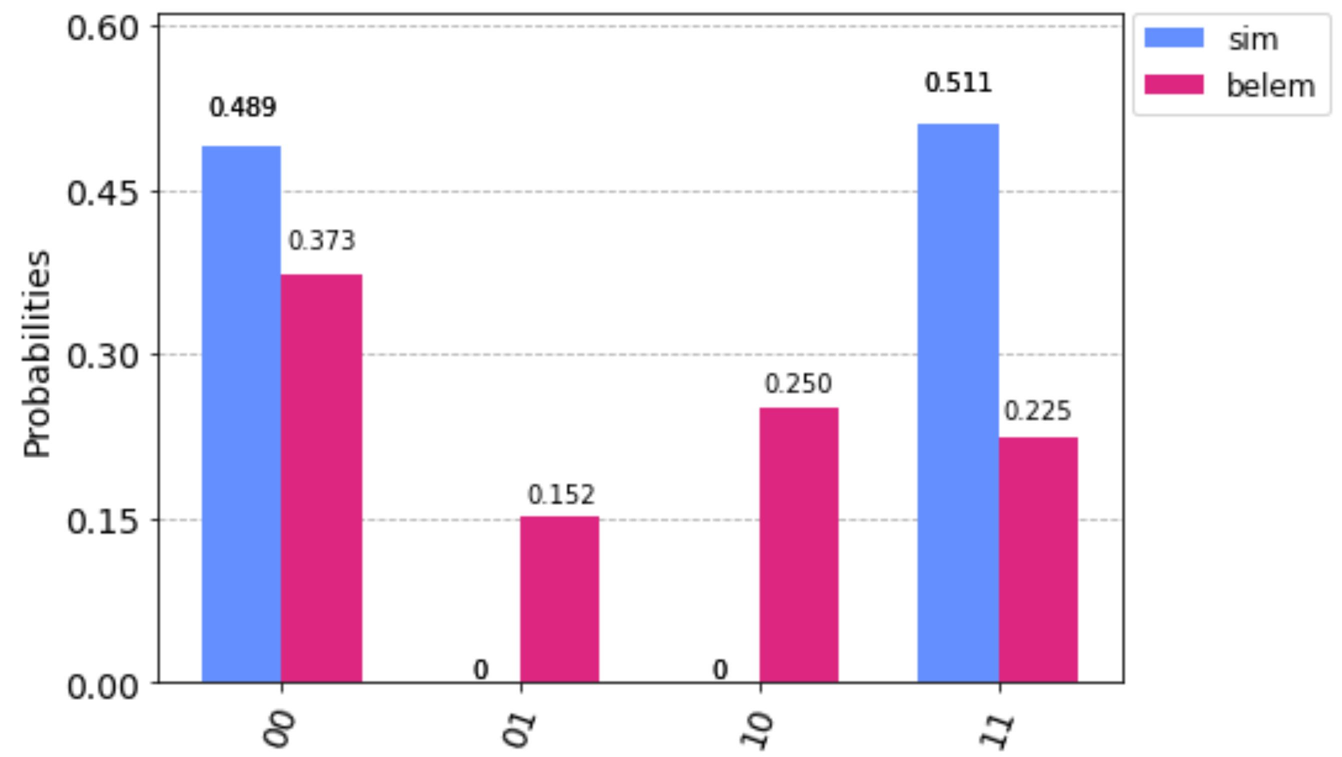




In [28]:
```
# Try this in a real device to see how it goes:
job_belem = execute(circ_s, backend=backend_belem)
job_monitor(job_belem)
result_belem = job_belem.result()
counts_belem = result_belem.get_counts(circ_s)
plot_histogram([counts_local,counts_belem], legend=['sim','belem'])
```

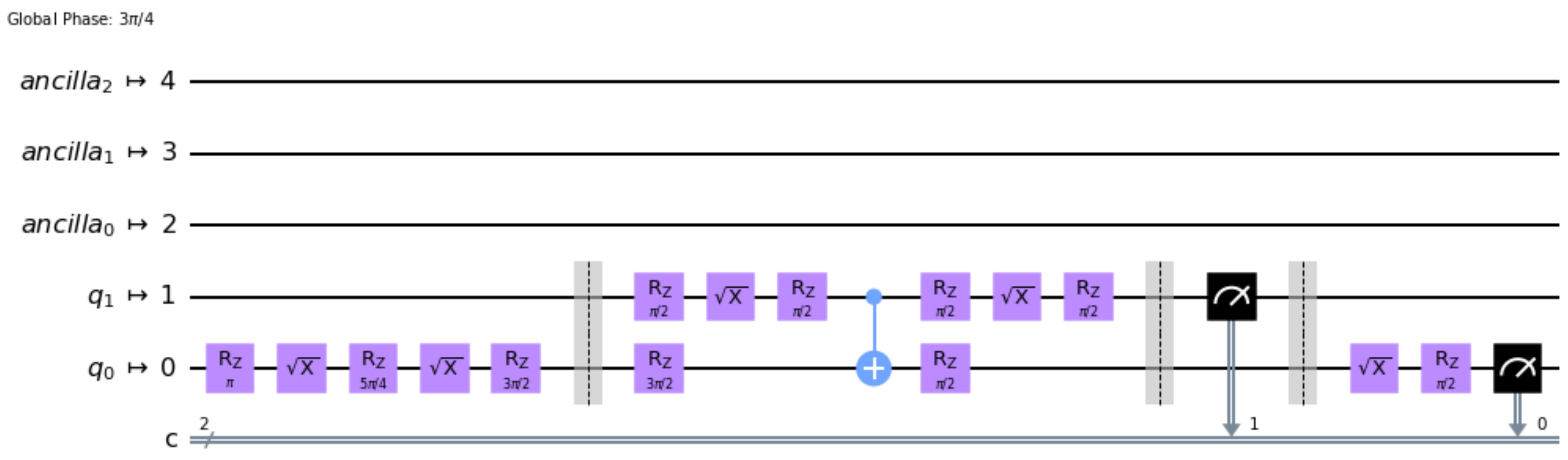
```
Job Status: job has successfully run
```

Out[28]:



In [37]:
```
# So the transpiled circuit looks quite complicated:
from qiskit import transpile
transpile(circ_s,backend_belem).draw(reverse_bits=True)
```

Out[37]:



In [ ]:
```
# The hardware does not have native H, S, and R_y gates, so these must be build from the \sqrt(X) and
R_z.
```

Adiabatic quantum computing

A "hook" to introduce Hamiltonians and Schröedinger's eqn.

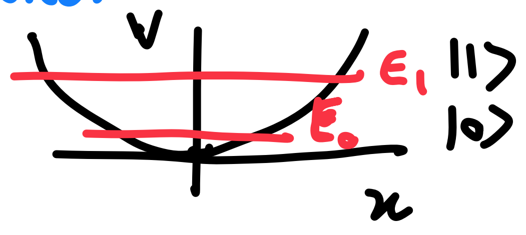
The Hamiltonian operator: Represents energy in quantum theory

- We learned that Hermitian operators describe "observables": things that we can measure. ENERGY is described by an operator called "Hamiltonian", denoted $\mathcal{H}$ (H in calligraphic font – not to be confused with Hadamard!).
- Example: Single qubit, like an "artificial atom".

$$\mathcal{H}_1 = \frac{\hbar\omega}{2}(I - Z) = \hbar\omega \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

TWO ENERGY LEVELS: $E_0 = 0$, $E_1 = \hbar\omega$

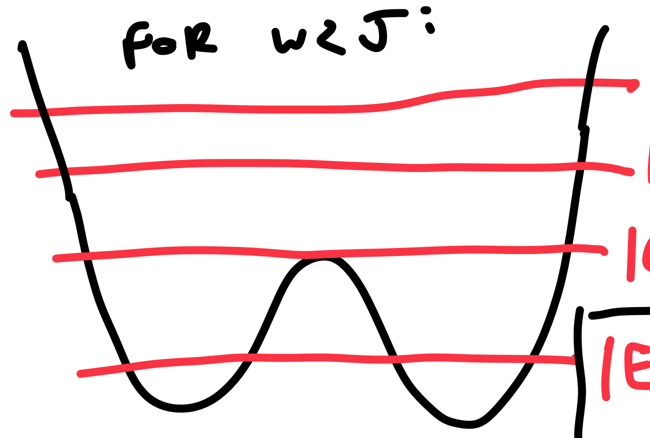
YOU GET THEM WHEN YOU MEASURE THE ENERGY OF THE SYSTEM: THE EIGENVALUES OF $\mathcal{H}$.



- Two qubits: Artificial molecule.

$$\mathcal{H}_2 = \begin{pmatrix} J/4 & 0 & 0 & 0 \\ 0 & (\omega - J/4) & J/2 & 0 \\ 0 & J/2 & (\omega - J/4) & 0 \\ 0 & 0 & 0 & (2\omega + J/4) \end{pmatrix}$$

FOR $\omega \ll J$:



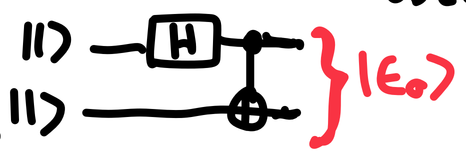
$|E_3\rangle = |11\rangle$

$|E_2\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$

$|E_1\rangle = |00\rangle$

$\boxed{|E_0\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)}$

TO GET THIS $|E_0\rangle$ USING CIRCUIT MODEL:



- The other states with higher energy are called *excited states*.

$\mathcal{H}(t)$ with <u>slow</u> time dependence

- But what if $\mathcal{H}$ changes in time? Sol. of Schröedinger's eqn is much more difficult. But at each time t we have "instantaneous eigenstates" $|E_j(t)\rangle$ satisfying

$$\boxed{\mathcal{H}(t)\,|E_j(t)\rangle = E_j(t)\,|E_j(t)\rangle}$$

- Assume $\mathcal{H}(t)$ changes very slowly (adiabatic) and the energy levels E_j do not coincide with each other. In this case, if at t=0 the qubits are in one of the eigenstates $|E_j(0)\rangle$ of $\mathcal{H}(t=0)$, they will remain in the "instantaneous eigenstate" $|E_j(t)\rangle$. Example:

$$\boxed{\mathcal{H}(t) = -\omega\,\{[1 - s(t)]\,Z + s(t)X\}}$$

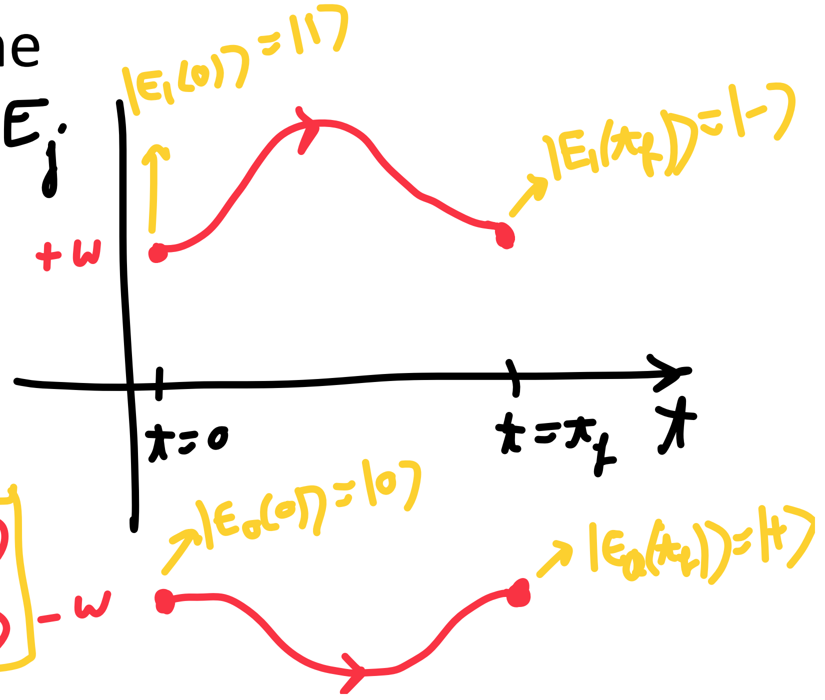
$s(t)$ IS CALLED "SCHEDULE" FNC.

$\begin{cases} s(t=0) = 0 \\ s(t=t_f) = 1 \end{cases}$ e.g. $s(t) = \frac{t}{t_f}$

AT $t=0$: $\mathcal{H}(t=0) = -\omega Z$

AT $t=t_f$: $\mathcal{H}(t_f) = -\omega X$

ADIABATIC HADAMARD: INPUT $|\psi(0)\rangle = |0\rangle \Rightarrow |\psi(t_f)\rangle = |+\rangle$; INPUT $|\psi(0)\rangle = |1\rangle \Rightarrow |\psi(t_f)\rangle = |-\rangle$



AQC for 3-satisfiability and quantum annealing as heuristic

The 3-satisfiability (3-SAT) problem

- 3-SAT: Find the n-bit string $z_{n-1}z_n \ldots z_0$ that satisfies a set of 3-bit constraint clauses. Each clause involves 3 bits i, j, k and the constraint is that one of the 3 bits must have value 1 and the other two must be 0:

$$c(i,j,k) = z_i + z_j + z_k - 1 = 0$$

AQC algorithm for 3-SAT E. Farhi et al, Science 292, 472 (2001)

READ THIS IF YOU ARE BORED (:) (IT'S A GREAT PAPER!)

a) Starting Hamiltonian:

$$\mathcal{H}(t=0) = \mathcal{H}(0) = \omega \sum_{i=0}^{n-1} \left(\frac{I - X_i}{2}\right)$$

PAULI-X ACTING ON QUBIT i.

$|E_0(0)\rangle = |+ \cdots +\rangle$

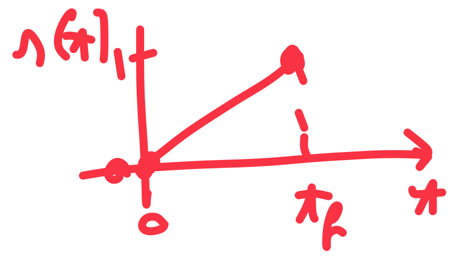
b) Final Hamiltonian:

$$\boxed{\mathcal{H}(t=t_f) = \mathcal{H}(1) = \sum_{\text{all } C(i,j,k)} \left[\left(\frac{I - Z_i}{2}\right) + \left(\frac{I - Z_j}{2}\right) + \left(\frac{I - Z_k}{2}\right) - 1\right]^2}$$

$\Rightarrow \mathcal{H}(1)|\text{STATE THAT SATISFIES ALL CLAUSES}\rangle = 0$!

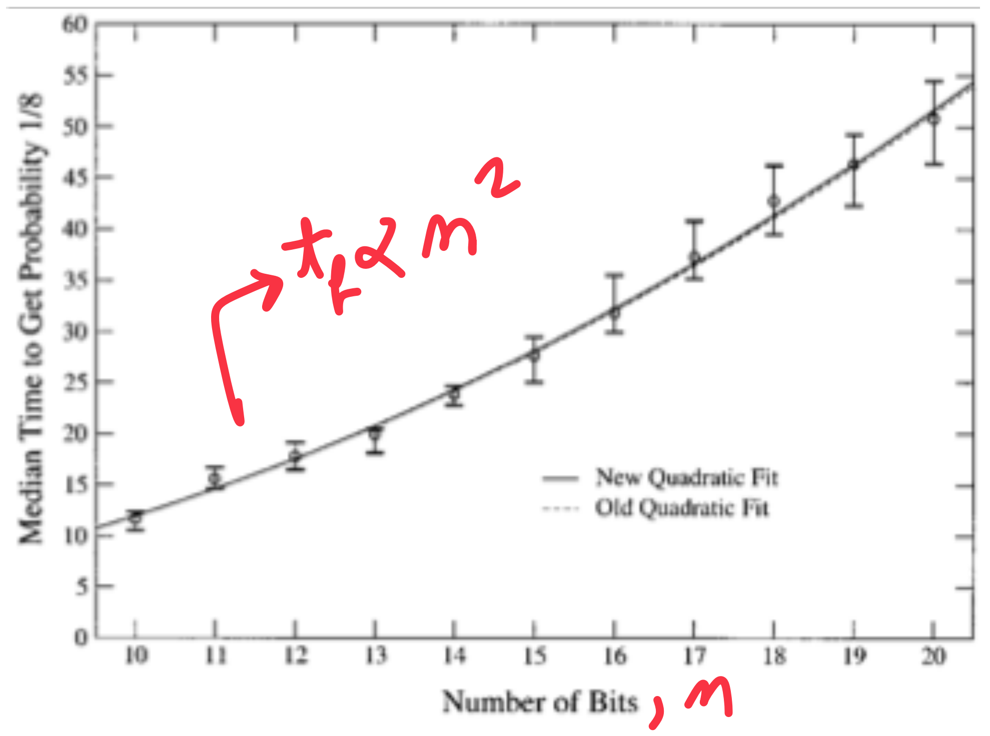
$\Rightarrow |E_0(1)\rangle$ IS ANSWER TO 3-SAT!

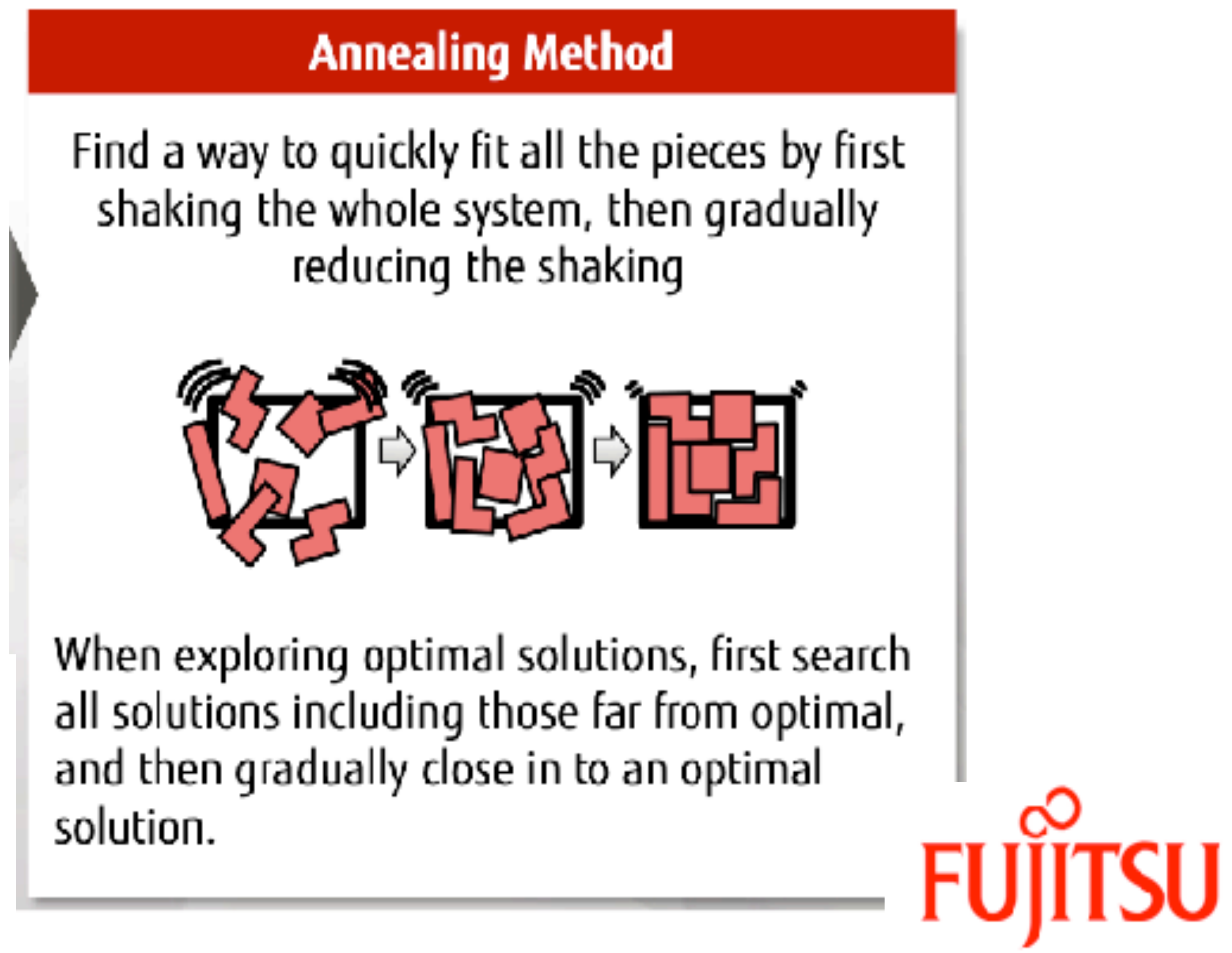
c) Picture shows median t_f required to get the right answer with prob. 1/8 using exact integration of Schröedinger eqn for linear s(t):



$s(t) = \frac{t}{t_f}$, CAN CHOOSE OTHER SCHEDULES!

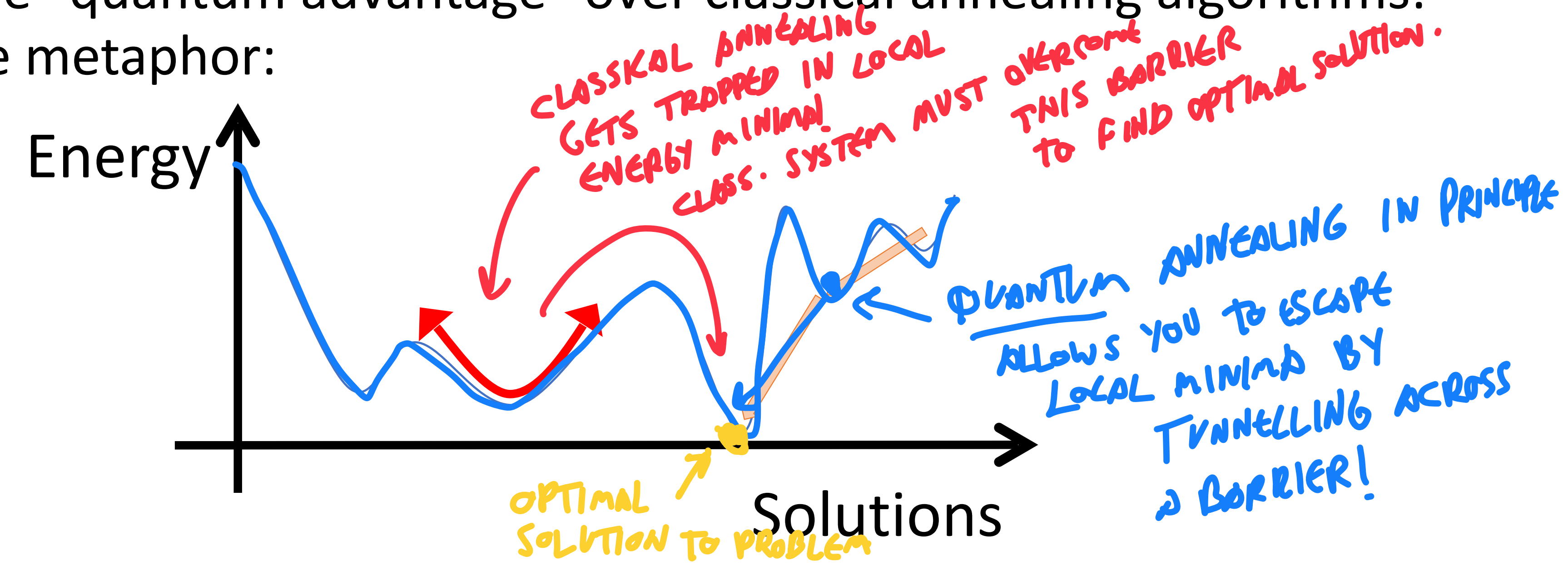
$$\boxed{\mathcal{H}(t) = \left(1 - \frac{t}{t_f}\right)\mathcal{H}(0) + \frac{t}{t_f}\mathcal{H}(1)}$$





Quantum annealing (QA) as heuristic AQC

- In Quantum annealing, one performs a **"loose run"** of the 3-SAT AQC for particular choices of s(t) and t_f, without worrying whether the system stays in the ground state or not. One experiments with s(t) and t_f with the goal of optimizing the probability of getting the right answer to the problem. There are no guarantees that QA works in all cases, it's trial and error (heuristics)!
- Why QA may have "quantum advantage" over classical annealing algorithms: Energy landscape metaphor:



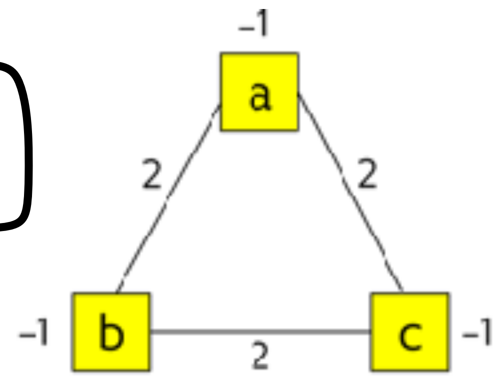
Experiential learning with D-Wave Leap

Learn to map problem to device architecture.

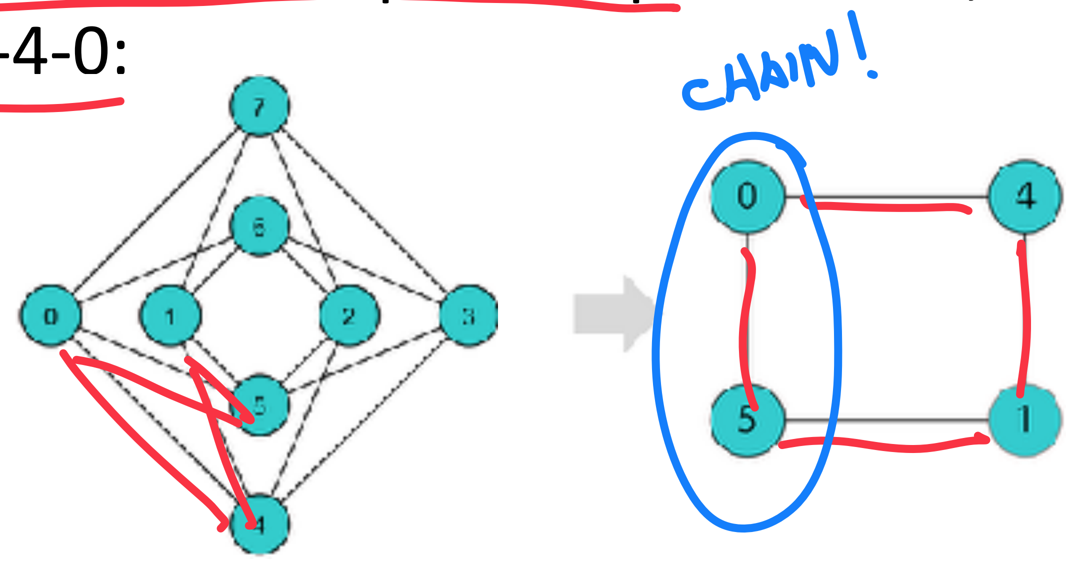
Minor-embedding: How to implement 3-qubit 3-SAT

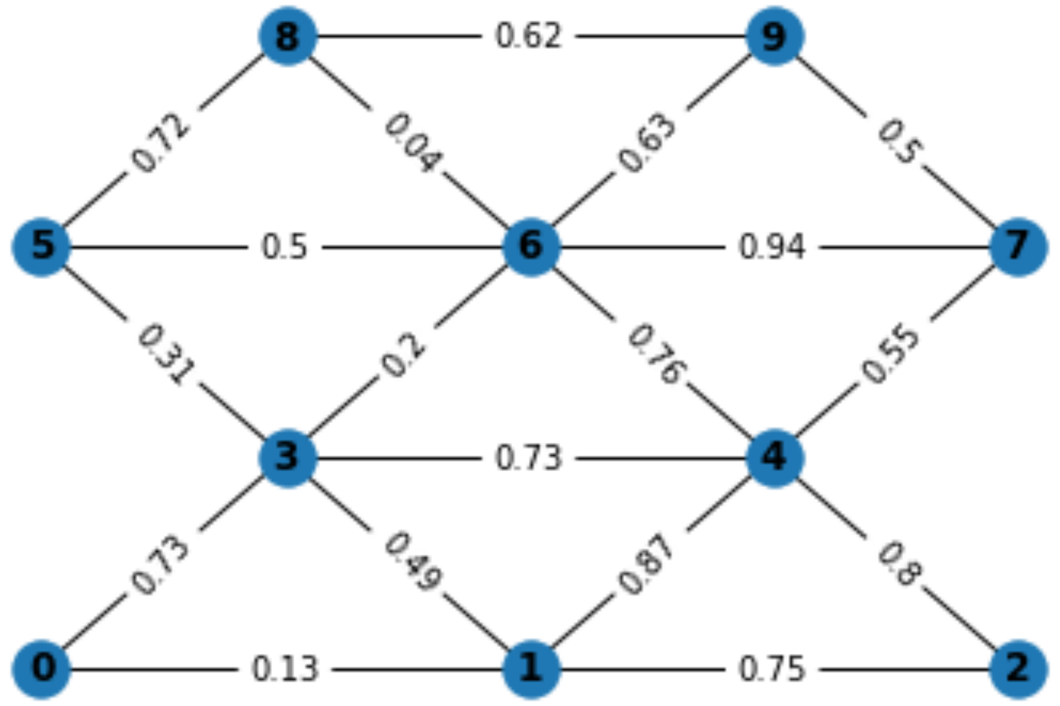
- To implement the simplest 3-SAT we need the QUBO:

$$\boxed{E(z) = (a+b+c-1)^2 = -(a+b+c) + 2\,(ab+ac+bc) + 1}$$



- Chimera does not have a closed 3-qubit loop. However, it does have a closed loop of 4 qubits: 0-5-1-4-0:





(a) Let's define the Ising problem by setting J=weights and h=0,0,... :

In [23]:
```
J = weights
h = dict()
#for i in G.nodes:
#    h.update({i: 0})
```

In [24]:
```
chainstrength = 2
numruns = 1000
# Run the QUBO on the solver from your config file
sampler = EmbeddingComposite(DWaveSampler())
response = sampler.sample_ising(h, J,
                                chain_strength=chainstrength,
                                num_reads=numruns)
```

In [28]:
```
# Print results
for sample, energy, num_occurrences, chain in response.data():
    print(sample, "Energy: ", energy, "Occurrences: ", num_occurrences)
```

```
{0: 1, 1: 1, 2: -1, 3: -1, 4: 1, 5: 1, 6: -1, 7: -1, 8: -1, 9: 1} Energy:  -5.909999999999999 Occurr
ences:  138
{0: -1, 1: -1, 2: 1, 3: 1, 4: -1, 5: -1, 6: 1, 7: 1, 8: 1, 9: -1} Energy:  -5.909999999999999 Occurr
ences:  327
{0: 1, 1: 1, 2: -1, 3: -1, 4: 1, 5: 1, 6: -1, 7: 1, 8: -1, 9: 1} Energy:  -5.69 Occurrences:  47
{0: -1, 1: -1, 2: 1, 3: 1, 4: -1, 5: -1, 6: 1, 7: -1, 8: 1, 9: -1} Energy:  -5.69 Occurrences:  185
{0: -1, 1: 1, 2: 1, 3: 1, 4: -1, 5: -1, 6: 1, 7: 1, 8: 1, 9: -1} Energy:  -5.43 Occurrences:  58
```

The best answer has E=-5.909999999999999 and the QPU finds this 138+327=465 out of 1000 runs (46.5%).

Final Exam

4. **(/10) Four-bit phase estimation.**– Generalize the phase estimation circuit in Problem 5 of Assignment 5 to four qubits, $j_1 j_2 j_3 j_4$ in the phase register. Implement in `Qiskit`.

   (a) Measure the phase of the operator $U = R_y(\theta)$ with $\frac{\theta}{4\pi}$ exactly represented by *four* base-2 decimals, $\frac{\theta}{4\pi} = 0.j_1 j_2 j_3 j_4$ (your choice of $j_1 j_2 j_3 j_4$). Note that $\frac{\theta}{4\pi}$ is the associated phase ϕ_u for $|u\rangle = |-i\rangle$, since

$$R_y(\theta)\,|-i\rangle = e^{+i\frac{\theta}{2}}\,|-i\rangle = e^{+2\pi i\frac{\theta}{4\pi}}\,|-i\rangle\,. \tag{4}$$

   Input $|\psi\rangle = |-i\rangle$, and run the circuit in the simulator and a real device to compare.

   (b) Run the circuit again for $\frac{\theta}{4\pi} = 0.j_1 j_2 j_3 j_4 - \epsilon$, with ϵ small. If you run the circuit many times does it return the $0.j_1 j_2 j_3 j_4$ closest to the correct answer $\frac{\theta}{4\pi}$? How does the real device compare to the simulator?

5. **(/8) Minimizing a quadratic polynomial with D-Wave Leap.**– Consider a four-bit representation for the real number $x \in [0, 1)$:

$$x = 0.j_1 j_2 j_3 j_4 = \frac{j_1}{2} + \frac{j_2}{2^2} + \frac{j_3}{2^3} + \frac{j_4}{2^4} = \frac{n}{16}, \quad \text{for } n = 0, 1, 2, \ldots, 15, \tag{5}$$

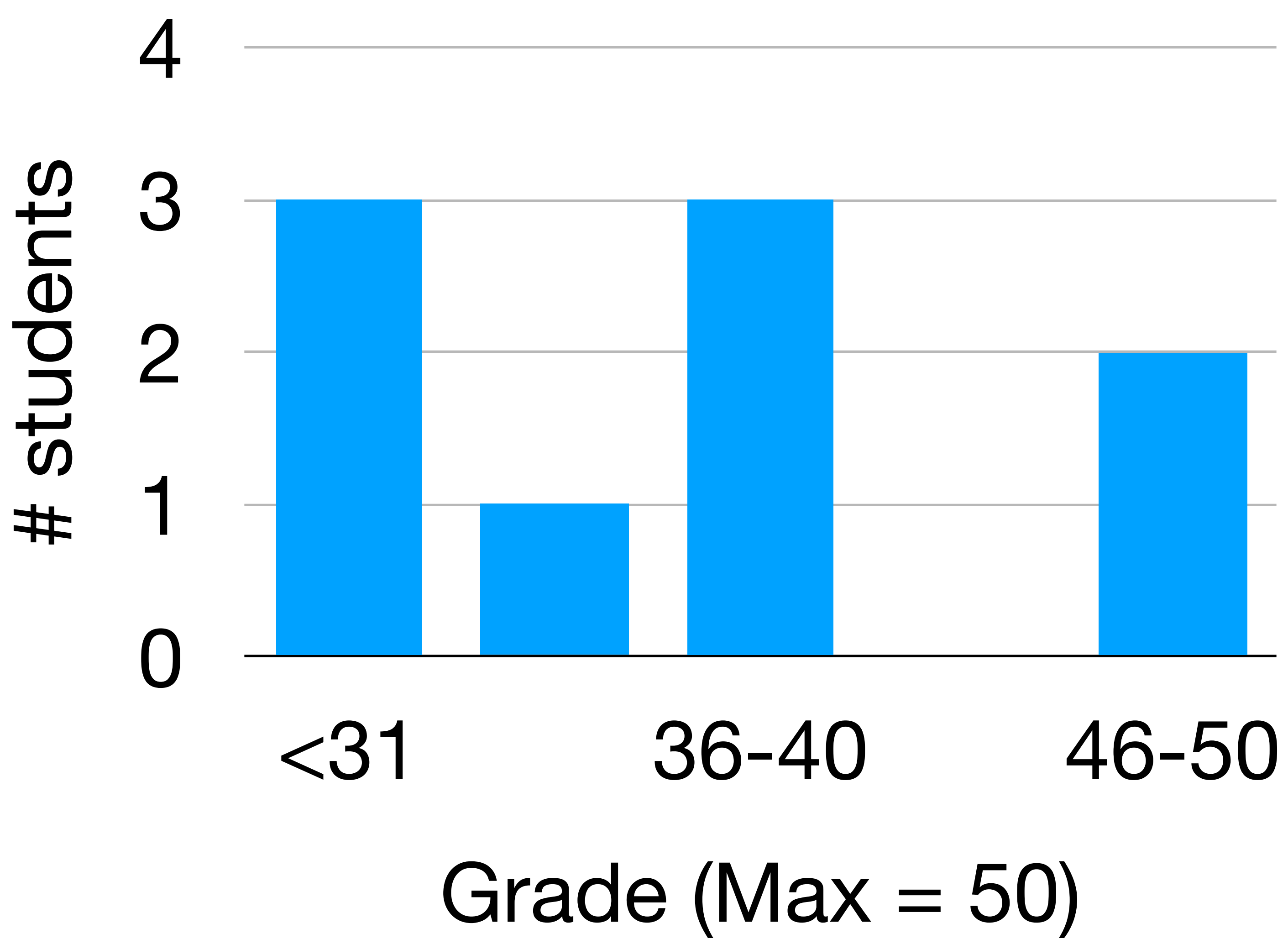
and the polynomial

$$p(x) = -x^2 + (0.94)x. \tag{6}$$

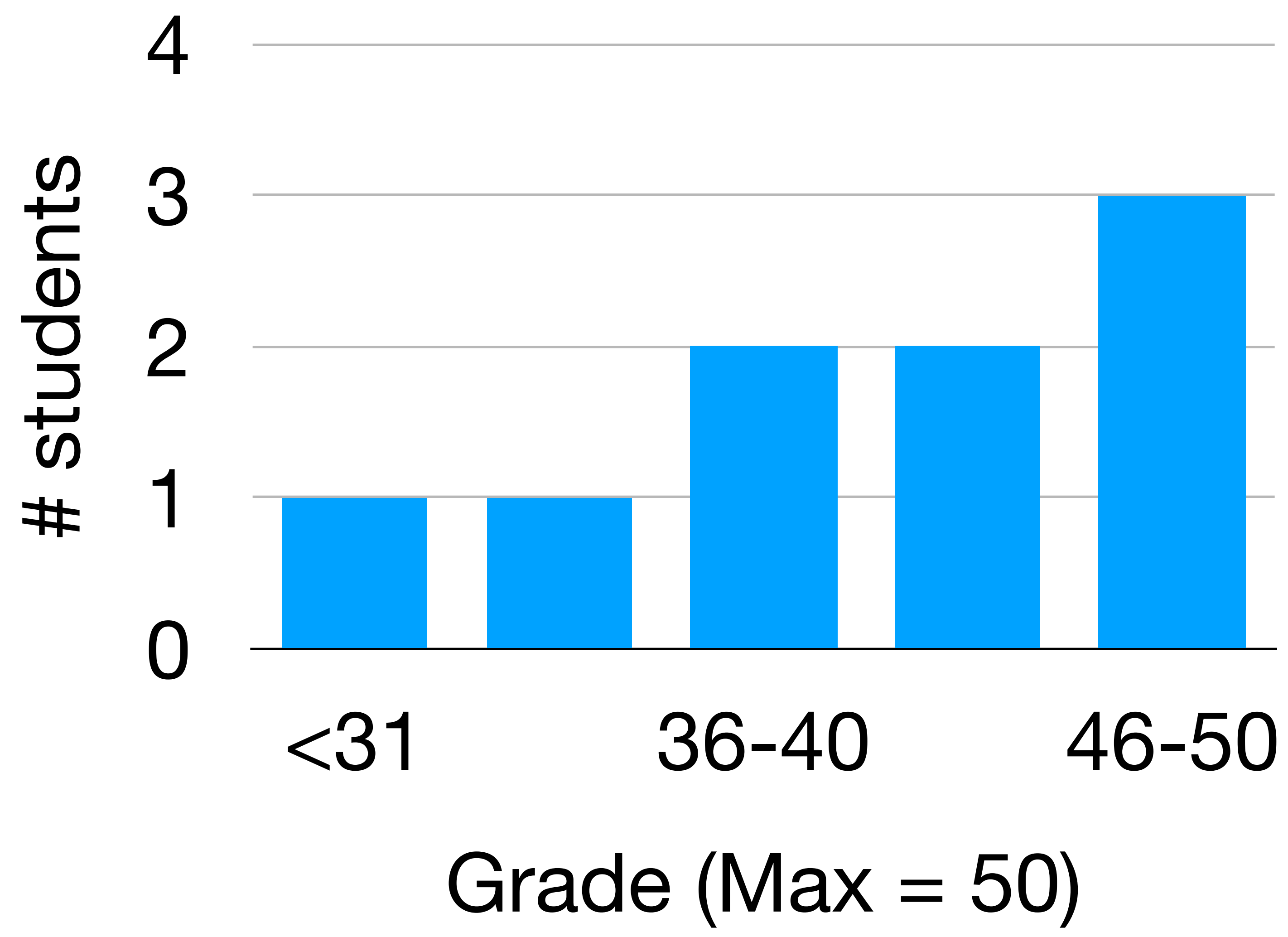
   (a) Use `matplotlib.pyplot` to plot $p(x)$ for the 16 allowed values of x. Locate the global and local minima.

   (b) Write an `Ocean-SDK` code that finds the x that minimizes the polynomial $p(x)$ in the QPU. Run 1000 times and determine the fraction of times the QPU finds the global minimum, the fraction of time it gets stuck in the local minimum, and the fraction of times it ends up in neither.

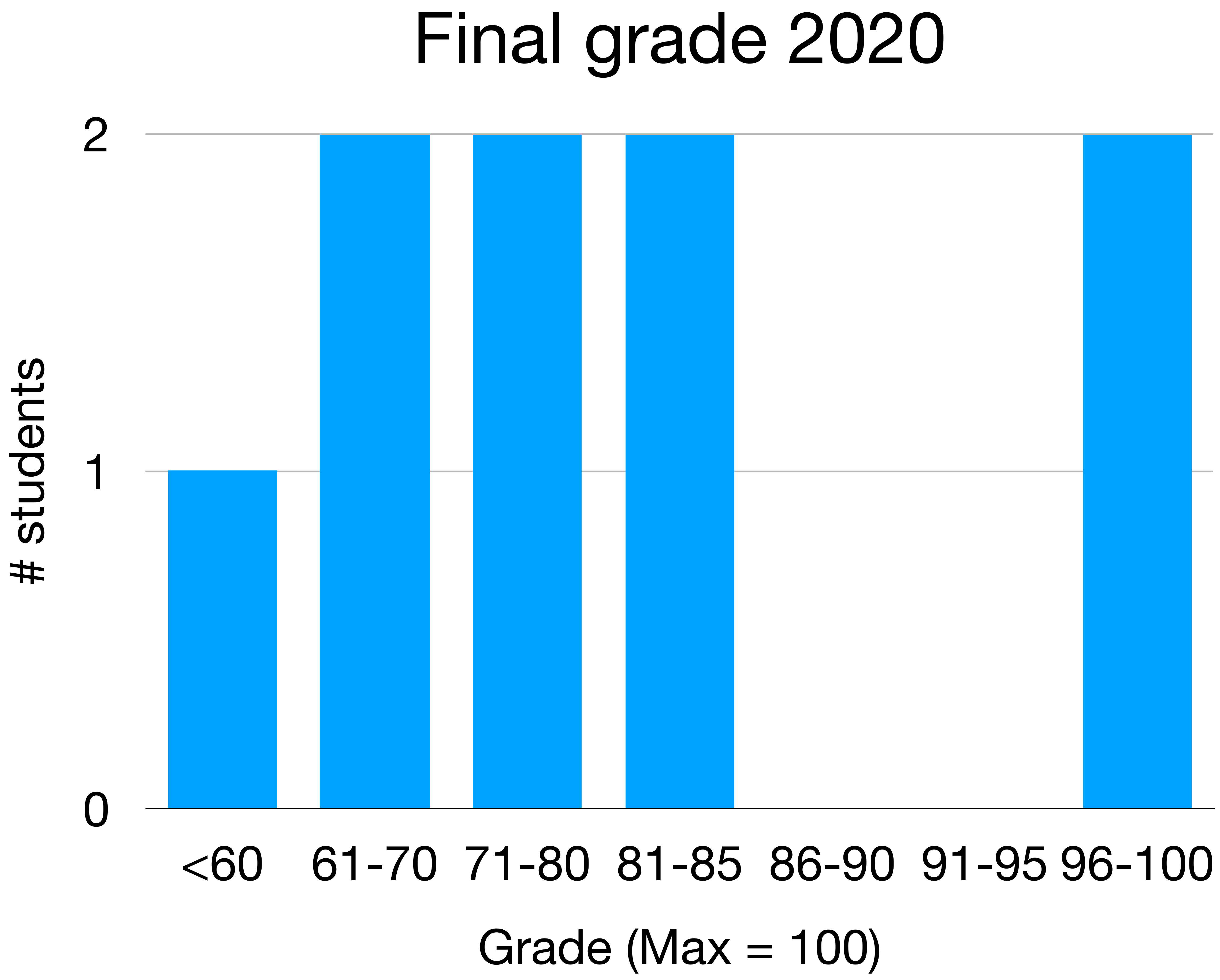
Final Exam 2020

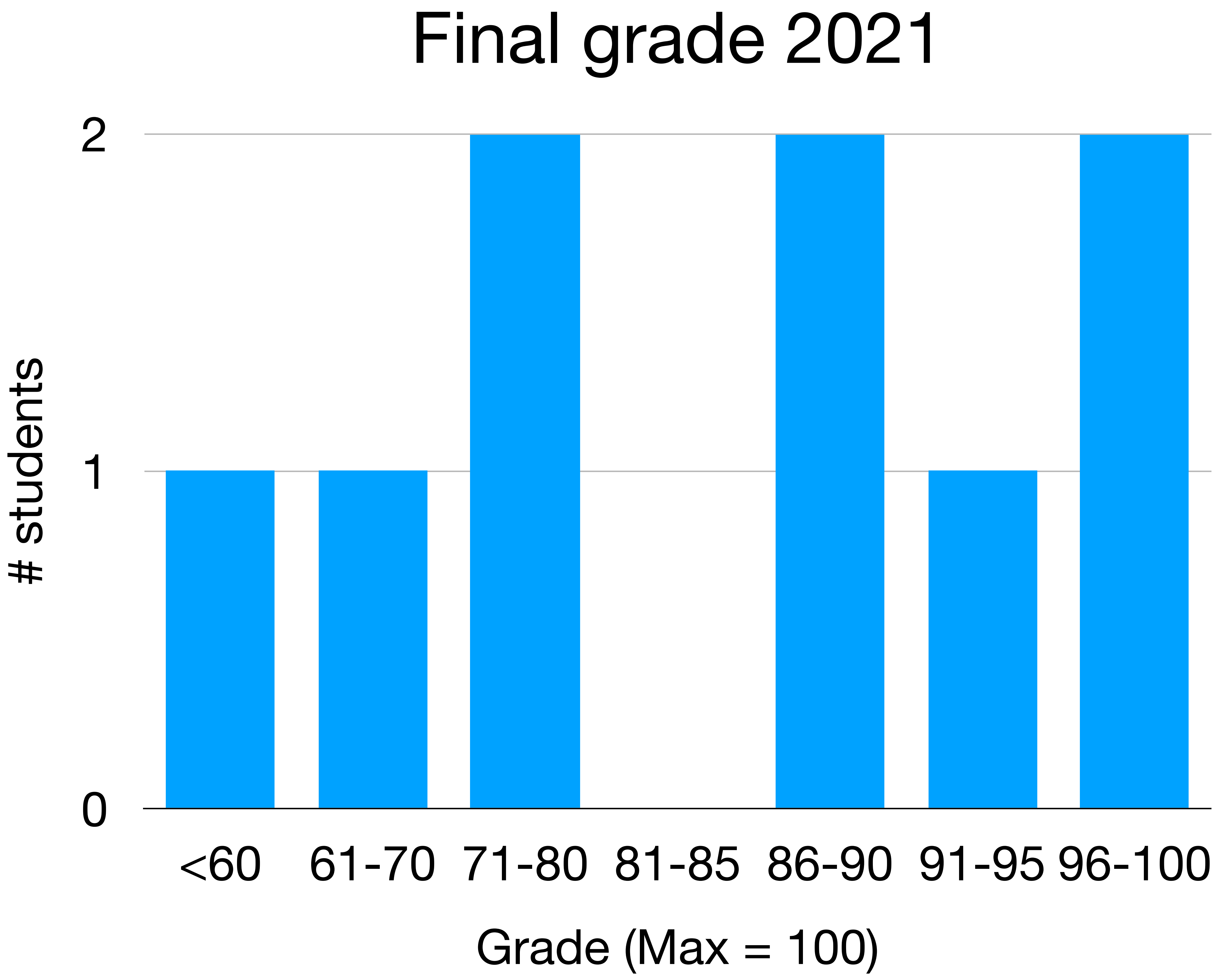


Final Exam 2021



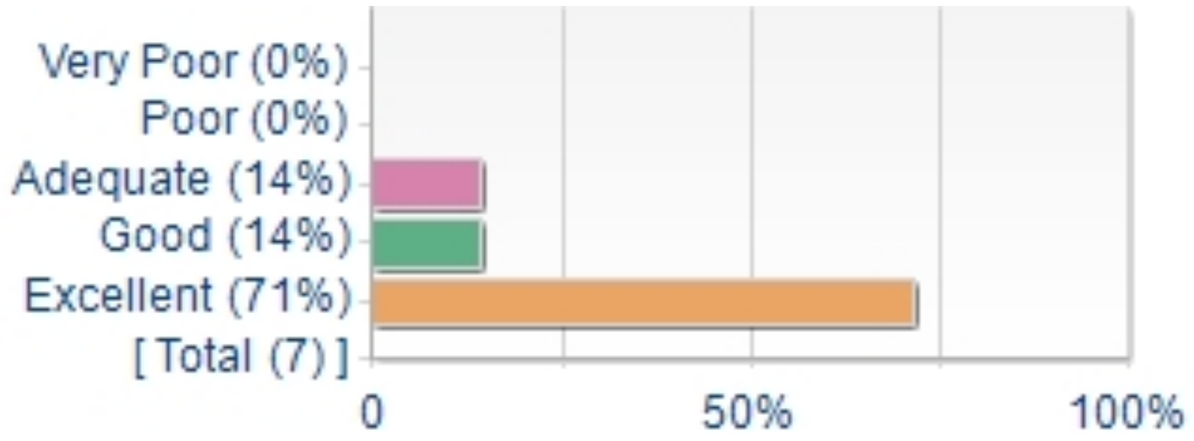
Final course grade



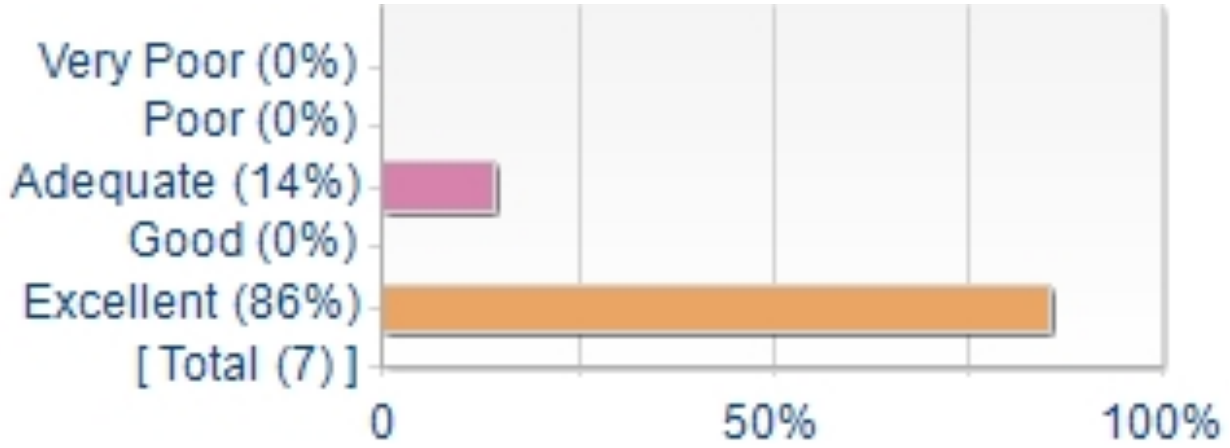


2020 Student feedback 2021

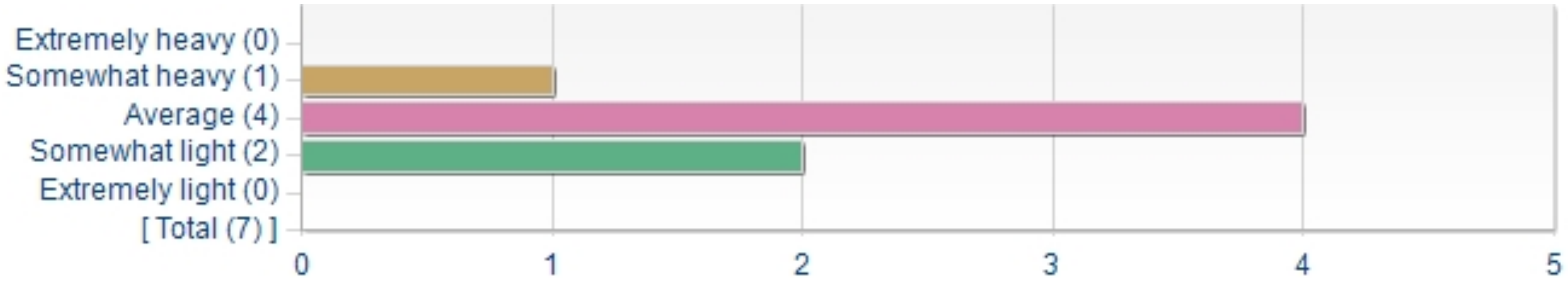
6. The course provided relevant skills and information (e.g. to other courses, your future career, or other contexts)



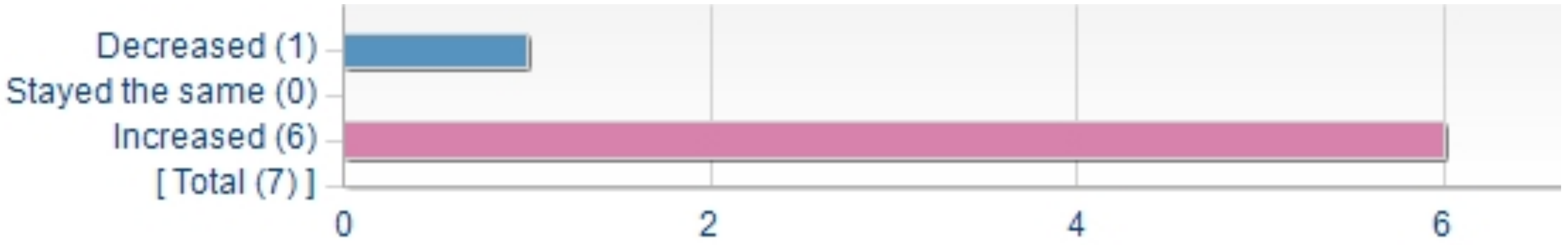
7. Overall, the course offered an effective learning experience



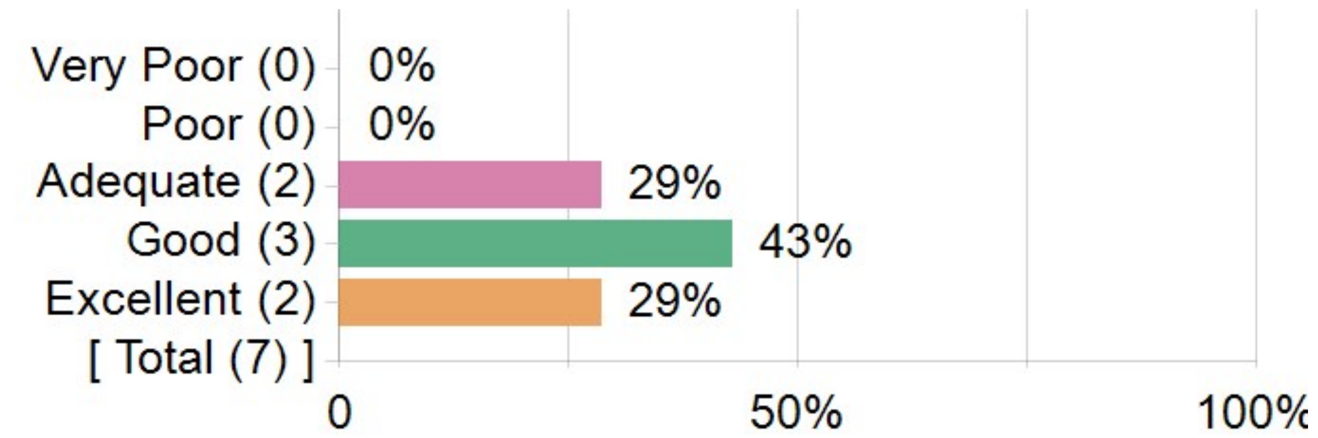
Relative to other courses I have taken at UVic, the workload in this course was



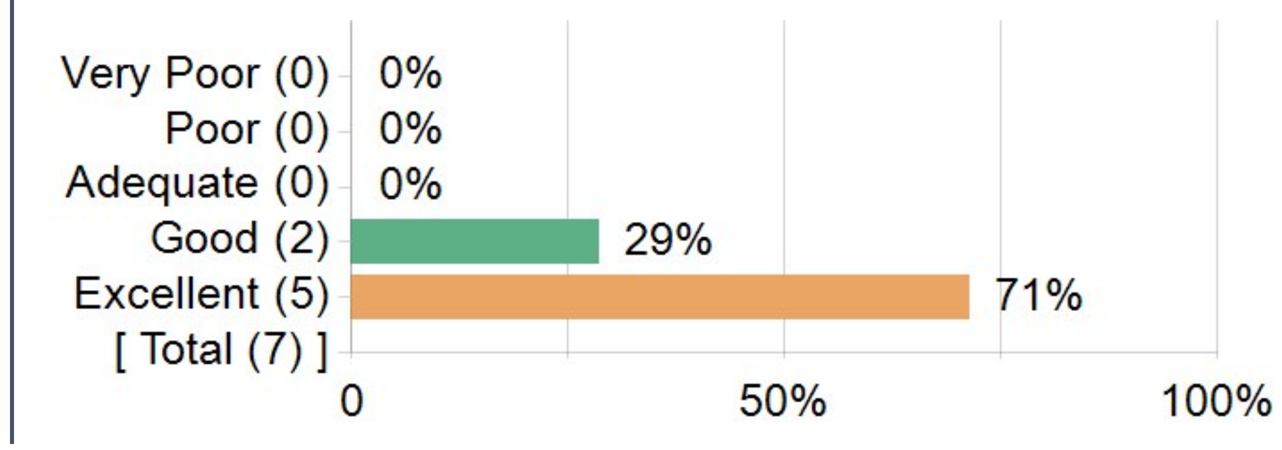
As a result of my experience in this course, my interest in the material:



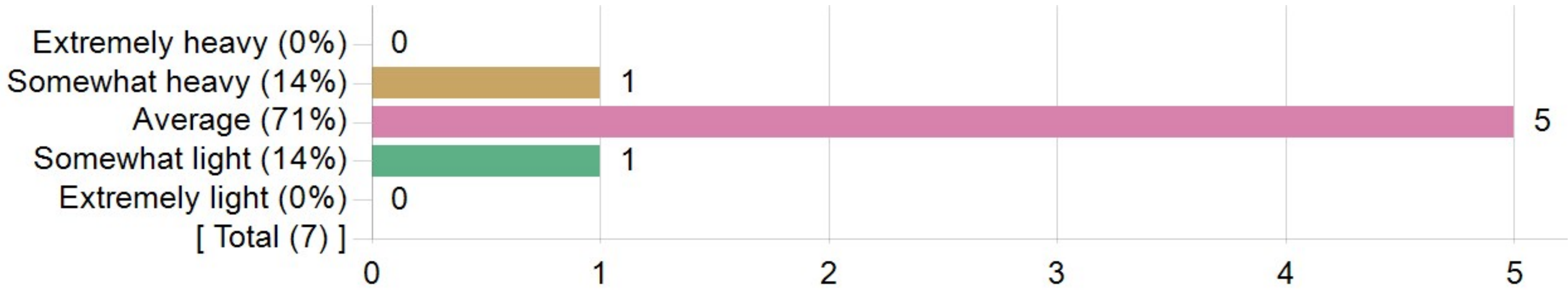
6. The course provided relevant skills and information (e.g. to other courses, your future career, or other contexts)



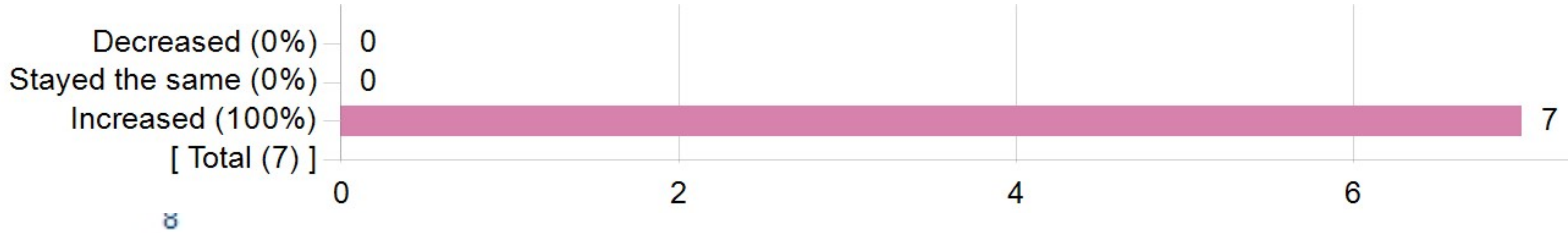
7. Overall, the course offered an effective learning experience



Relative to other courses I have taken at UVic, the workload in this course was



As a result of my experience in this course, my interest in the material:



Student feedback: Two representative comments

2020:

“Having a small class, and allotting time in class to work through problems with other students, as well as on the IBMQ and D–Wave systems was very appreciated! My interest in quantum computing has soared after taking this class, and I think it will be interesting to relate what I have learned to other computer science and physics courses.”

2021:

“Group work is an excellent way to facilitate learning, but it seems to be less effective with such small group of students (sometimes only two). In a course this size, I might suggest letting the entire class work together on a problem to achieve a good group dynamic with more frequent exchanges of ideas.”

Conclusions, open questions

- “Gentle" approach for teaching QC to 2nd year students with only linear algebra as pre-req. Did it work? Need help from Phys. Education researchers to give a quantum proficiency test!

- Allotting time in-class for students to work on problems as a group seemed to work quite well. Lots of opportunity to do this with simple theory problems, IBM-Q, D-Wave Leap coding, etc.

- I felt that “hands-on” with IBM-Q and D-Wave Leap was useful to crystallize concepts and formalism. Requires effort to design assignments and exams, e.g. quantum phase estimation without Q Fourier transform. How to quantify the benefit of NISQ devices on Q education?

- Teaching adiabatic QC opens student’s minds to QC models other than gate model, and is a great hook for teaching Hamiltonians and Schröedinger’s eqn. Hands-on with D-Wave Leap reinforced the concept of Hamiltonians, ground state, and problem embedding into Hamiltonian that fits device architecture.

- Can "Intro to QC” for 2nd years be used as an alternative to the usual “historic intro” to quantum theory?

- Can we get rid of linear algebra pre-req (Would allow e.g. Chemistry students to take the course).