QALCO: Quantum Algorithms and Complexity **2025 - 2026**

Lecture 5: BQP, BQP vs. NP, QMA

Lecturer: Alex Bredariol Grilo (alex.bredariol-grilo@lip6.fr)

1 Recap

1.1 P, BPP, PSPACE

Definition 1 (P)**.** $(L_{yes}, L_{no}) \in \mathsf{P}$ *iff there exists a polynomial p and a Turing Machine M such that on input $x \in L_{yes} \cup L_{no}$, $M(x)$ runs in $p(|x|)$-time and*

completeness: if $x \in L_{yes}$, then $M(x)$ accepts

soundness: if $x \in L_{no}$, then $M(x)$ rejects.

Definition 2 (BPP)**.** $(L_{yes}, L_{no}) \in \mathsf{BPP}$ *iff there exists polynomials p and q and a Turing Machine M such that on input $x \in L_{yes} \cup L_{no}$ and $r \in \{0,1\}^{q(|x|)}$, $M(x,r)$ runs in $p(|x|)$-time and*

completeness: if $x \in L_{yes}$, then $\Pr_{r\in\{0,1\}^{q(|x|)}}[M(x,r) \text{ accepts }] \geq \frac{2}{3}$;

soundness: if $x \in L_{no}$, then $\Pr_{r\in\{0,1\}^{q(|x|)}}[M(x,r) \text{ accepts }] \leq \frac{1}{3}$.

Definition 3 (PSPACE)**.** $(L_{yes}, L_{no}) \in \mathsf{PSPACE}$ *iff there exists a polynomial p and a Turing Machine M such that on input $x \in L_{yes} \cup L_{no}$, $M(x)$ uses at most $p(|x|)$ space* [1] *and*

completeness: if $x \in L_{yes}$, then $M(x)$ accepts

soundness: if $x \in L_{no}$, then $M(x)$ rejects.

2 Quantum complexity theory

Switching gears to quantum computing, we can define BQP as the set of problems that can be solved in quantum polynomial time.

Definition 4 (BQP)**.** $(L_{yes}, L_{no}) \in \mathsf{BQP}$ *iff there exists polynomials p and q and a uniform family of quantum circuits $\{C_n\}_{n\in\mathbb{N}}$, where C_n acts on $n + q(n)$ qubits and it is composed of $p(n)$ gates such that*

completeness: if $x \in L_{yes}$, then $||(|1\rangle\langle 1| \otimes I)C_{|x|}\,|x\rangle\,|0\rangle^{\otimes q(|x|)}\,||^2 \geq \frac{2}{3}$;

soundness: if $x \in L_{no}$, then $||(|1\rangle\langle 1| \otimes I)C_{|x|}\,|x\rangle\,|0\rangle^{\otimes q(|x|)}\,||^2 \leq \frac{1}{3}$;

Remark 1. *We could define* BQP *with quantum Turing Machines but it is much more complicated.*

Theorem 1 ([3])**.** $(\mathsf{FACT}_{yes}, \mathsf{FACT}_{no}) \in \mathsf{BQP}$.

We could have a more general definition $\mathsf{BQP}(c,s)$ and consider the completeness parameter c (instead of $\frac{2}{3}$) and soundness parameter s (instead of $\frac{1}{3}$). The following lemma shows that this actually does not change the computational power of the complexity class for reasonable choices of c and s, as long as there is an inverse polynomial gap $c - s = \frac{1}{poly(n)}$.

[1] Think of it as the memory used by an algorithm.

Theorem 2. *For every polynomial p and negligible function η, $\mathsf{BQP}(c, c - \frac{1}{p}) = \mathsf{BQP}(1-\eta, \eta)$.*

Proof sketch. Let $(L_{yes}, L_{no}) \in \mathsf{BQP}(c, c - \frac{1}{p})$ and let $\{C_n\}_{n\in\mathbb{N}}$ be corresponding uniform family of circuits from Definition 4. We can define a uniform family $\{C'_n\}_{n\in\mathbb{N}}$, where C'_n

1. Runs C_n $t = p(|x|)^2$ times
2. Accept if $t(c - \frac{1}{2p(|x|)})$ runs of M accept
3. Reject otherwise.

Let X_i be the random variable s.t. $X_i = 1$ is the i-th run of C_n accepts, and $X_i = 0$ otherwise. If $x \in L_{yes}$, we have from Chernoff bounds (see Appendix A) that

$$\Pr[M'(x) rejects] = \Pr\left[\sum_i X_i \leq t(c - \frac{1}{2p(|x|)})\right] \leq 2^{-O(p(n))}. \tag{1}$$

If $x \in L_{yes}$, we have that

$$\Pr[M'(x) rejects] = \Pr\left[\sum_i X_i \geq t(c - \frac{1}{2p(|x|)})\right] \leq 2^{-O(p(n))}. \tag{2}$$

You will do the full calculations in a similar exercise. □

2.1 Known inclusions

Theorem 3. $\mathsf{BPP} \subseteq \mathsf{BQP}$.

Proof sketch. First, notice that $\mathsf{P} \subseteq \mathsf{BQP}$. This can be shown by noticing that the set of gates $\{\mathsf{Toffoli}, \mathsf{X}\}$ is universal for classical computation and both gates are unitary. In this case, we can use the equivalence between Turing Machines and uniform family of quantum circuits to prove this inclusion.

To lift this result to $\mathsf{BPP} \subseteq \mathsf{BQP}$, we can notice that $H|0\rangle = |+\rangle$ and it can be used to simulate the coins needed for BPP. In this case, we can simulate the TM that solve a BPP problem on input $\frac{1}{\sqrt{2^{q(|x|)}}} |x\rangle \sum_r |r\rangle$. □

Definition 5 (PSPACE)**.** $(L_{yes}, L_{no}) \in \mathsf{PSPACE}$ *iff there exists a polynomial p and a Turing Machine M such that on input $x \in L_{yes} \cup L_{no}$, $M(x)$ uses at most $p(|x|)$ space* [2] *and*

completeness: if $x \in L_{yes}$, then $M(x)$ accepts

soundness: if $x \in L_{no}$, then $M(x)$ rejects.

Notice that a problem in PSPACE could potentially take exponential time. We know that $\mathsf{BPP} \subseteq \mathsf{PSPACE}$, and we believe that this containment is strict, but again, this is an open problem in complexity theory. We will now show that $\mathsf{BQP} \subseteq \mathsf{PSPACE}$.

Theorem 4. $\mathsf{BQP} \subseteq \mathsf{PSPACE}$.

[2] Think of it as the memory used by an algorithm.

Proof. Let us consider a circuit $U = U_T ... U_1$ acting on m qubits and composed of gates in the set $\{\mathsf{H}, \mathsf{Toff}, \mathsf{X}\}$. We notice that this gateset is universal for quantum computation.

We have that for all $y_0, y_T \in \{0,1\}^m$,

$$\begin{aligned} \langle y_T | U_T....U_1 |y_0\rangle &= \langle y_T | U_T I U_{T_1}....U_2 I U_1 |y_0\rangle && (3) \\ &= \sum_{y_1,\ldots,y_{T-1}} \langle y_T | U_T |y_{T-1}\rangle\langle y_{T-1}| U_{T_1}.... |y_1\rangle\langle y_1| I U_1 |y_0\rangle && (4) \\ &= \sum_{y_1,\ldots,y_{T-1}} \prod_{i=0}^{T-1} \langle y_{i+1}| U_{i+1} |y_i\rangle && (5) \end{aligned}$$

Notice that $\langle y_{i+1}| U_{i+1} |y_i\rangle$ is an entry of the matric U_{i+1} and it can be computed in polynomial time. In this case, $\prod_{i=0}^{T-1} \langle y_{i+1}| U_{i+1} |y_i\rangle$ cane be computed in polynomial time for fixed $y_0, ..., y_T$. Finally, $\sum_{y_1,\ldots,y_{T-1}} \prod_{i=0}^{T-1} \langle y_{i+1}| U_{i+1} |y_i\rangle$ can be computed in polynomial space (Exercise!).

To finish the proof, to compute the probability that U outputs 1, we need to compute the value

$$\sum_{y_T \in \{1\}\times\{0,1\}^{m-1}} \left|\langle y_T | U_T....U_1 |x\rangle |0^q\rangle\right|^2 . \qquad (6)$$

□

3 NP, NP-hardness, NP-completeness

Another important complexity class is NP, which models efficient solution verification.

Definition 6. $(L_{yes}, L_{no}) \in \mathsf{NP}$ *iff there exist a polynomial p and a polynomial-time algorithm V* [3]*, such that*

completeness: if $x \in L_{yes}$, then $\exists y \in \{0,1\}^{p(|x|)}$ such that $V(x,y) = 1$;

soundness: if $x \in L_{no}$, then $\forall y \in \{0,1\}^{p(|x|)}$, $V(x,y) = 0$.

Example of problems in NP: 3SAT, 3-Coloring, TSP, Ising model

We will define an important subset of NP, which are the class of problems that are *formally* harder than any problem in NP. For that, we need to introduce the notion of reduction.

Definition 7. *We say that a problem $L_1 = (L^1_{yes}, L^1_{no})$ reduces to $L_2 = (L^2_{yes}, L^2_{no})$ ($L_1 \leq L_2$), if there is a polynomial-time (classical) algorithm A such that*

- *if $x \in L^1_{yes}$, then $A(x) \in L^2_{yes}$, and*
- *if $x \in L^1_{no}$, then $A(x) \in L^2_{no}$.*

This notion of reduction allows us to compare how hard is problem: if $L_1 \leq L_2$, then a polynomial time algorithm for L_2 gives you a polynomial algorithm for L_1. Or in the contrapositive way, if we know that L_1 cannot be solved in polynomial time, then L_2 cannot be solved in polynomial time either.

With this notion in hand, we can define NP-problems.

Definition 8. *A problem $L = (L_{yes}, L_{no})$ is NP-hard if for every problem $L' = (L'_{yes}, L'_{no}) \in \mathsf{NP}$, $L' \leq L$.*

[3] From now on, I am using this shortcut for Turing Machines that run in time $q(|x|)$ for some polynomial q. Moreover we say that the TM accepts if the algorithm outputs 1 and it rejects if the algorithm outputs 0.

Definition 9. *A problem $L = (L_{yes}, L_{no})$ is NP-complete iff it is NP-hard and in NP.*

In this case, NP-complete problems are the "hardest" problems in NP.

We trivially have that $\mathsf{P} \subseteq \mathsf{NP}$, but showing if such inclusion is strict or if $\mathsf{P} = \mathsf{NP}$ is the most important open problem in complexity theory (and it is one of the Millenium problems of the Clay Institute). Notice that by definition, if we have a polynomial-time algorithm for *any* NP-hard problem, then $\mathsf{P} = \mathsf{NP}$.

4 Oracle separations

We believe that BQP and NP are uncomparable (i.e. $\mathsf{BQP} \not\subseteq \mathsf{NP}$ and $\mathsf{NP} \not\subseteq \mathsf{BQP}$), but proving such a statement is beyond our current techniques. Instead, we have ways of gaining some intuition as to why this should be true and also get some ideas of proofs that could or could not work.

One way of doing this is working with oracles. In a complexity class C with oracle access to a function O, denoted by C^O, the algorithms in C can make query O on an input x and they get back the answer $O(x)$. To deal with quantum oracles, we actually consider a (non-uniform) family of oracles $O = \{O_n\}$, and a quantum query with an arbitrary state $\sum_{x,b} \alpha_{x,b} |x\rangle |b\rangle$ has the outcome of the oracle call is $\sum_{x,b} \alpha_{x,b} |x\rangle |b \oplus O_n(x)\rangle$.

Thus, even if we don't know how to show that $\mathsf{C}_1 \not\subseteq \mathsf{C}_2$, sometimes we can show that we have an oracle O such that $\mathsf{C}_1^O \not\subseteq \mathsf{C}_2^O$. A proof of the latter statement heavily relies that we have only oracle access to O and we don't know, for example, a circuit to implement it. On the one hand, such statements are useful because they show that a proof $\mathsf{C}_1 \subseteq \mathsf{C}_2$ should fail if we add oracles (i.e., the proof does not relativize). On the other hand, very important theorems in complexity theory do not relativize (e.g. PCP theorem or $\mathsf{IP} = \mathsf{PSPACE}$). Therefore, such oracle separations are important, but they should be taken with a grain of salt.

Theorem 5 ([1])**.** *There exists an oracle such that*

$$\mathsf{NP}^O \not\subseteq \mathsf{BQP}^O.$$

Proof sketch. Let $O = \{O_n\}$ be a fixed family of oracles. We have define the language

$$L_O = \{1^n : \exists x \in \{0,1\}^n \text{ s.t. } O_n(x) = 1\}. \tag{7}$$

This problem is trivially in NP^O: on input 1^n the verifier can receive a candidate $x \in \{0,1\}^n$ and check if $O_n(x) = 1$. If we have a positive instance, there exists an x that satisfies it, whereas if it is a no-case, all possible values will make the verifier reject.

On the other hand, if we only have oracle access to O, the optimality of Grover's algorithm to find a marked element shows that no algorithm that makes a polynomial number of queries to O can distinguish the two cases. Therefore, this problem cannot be in BQP^O. □

Theorem 6 ([2])**.** *There exists an oracle such that*

$$\mathsf{BQP}^O \not\subseteq \mathsf{NP}^O.$$

The proof of this theorem is rather complicated, and actually it shows that $\mathsf{BQP}^O \subsetneq \mathsf{PH}^O$, where PH is a generalization of NP.

5 QMA and the local Hamiltonian problem

Before going to quantum verification of proofs, let us generalize NP to the randomized setting.

Definition 10 (MA)**.** $(L_{yes}, L_{no}) \in \mathsf{MA}$ *iff there exist polynomials p and q, and a polynomial-time algorithm V, such that*

> *complet.: if* $x \in L_{yes}$*, then* $\exists y \in \{0,1\}^{p(|x|)}$ *such that* $\Pr_{r \in \{0,1\}^{q(|x|)}}[V(x,y,r) \text{ accepts }] \geq \frac{2}{3}$;
>
> *sound.: if* $x \in L_{no}$*, then* $\forall y \in \{0,1\}^{p(|x|)}$, $\Pr_{r \in \{0,1\}^{q(|x|)}}[V(x,y,r) \text{ accepts }] \geq \frac{2}{3}$;

It is easy to see that NP ⊆ MA. While we believe that NP = MA (in fact, this is implied by the conjecture that P = BPP), this is an open problem. Moreover, the NP vs. MA problem has tight connections to Hamiltonian complexity which is covered below.

One interesting characteristic about MA is that we can make it *perfect complete*, i.e., for every problem in MA, there is a verification algorithm that always accepts yes-instances.

Finally, we notice that there are other randomized generalizations of NP. For example, the AM is defined very similarly to MA, but the string y can depend on the randomness r. We have that MA ⊆ AM, but we also find it plausible that AM = NP.

We will now switch gears to the quantum analog of NP/MA.

Definition 11 (QMA)**.** $(L_{yes}, L_{no}) \in \mathsf{QMA}$ *iff there exist polynomials p and q, and a uniform family of polynomial-time circuits* $\{V_n\}_n$ *such that*

> *completeness: if* $x \in L_{yes}$*, then there exists a* $p(|x|)$*-qubit state* $|\psi\rangle$ *such that*
>
> $$||(|1\rangle\langle 1| \otimes I)C_{|x|}\,|x\rangle\,|\psi\rangle\,|0\rangle^{\otimes q(|x|)}\,||^2 \geq \frac{2}{3};$$
>
> *soundness: if* $x \in L_{no}$*, then for every* $p(|x|)$*-qubit* $|\psi\rangle$, $||(|1\rangle\langle 1| \otimes I)C_{|x|}\,|x\rangle\,|\psi\rangle\,|0\rangle^{\otimes q(|x|)}\,||^2 \leq \frac{1}{3}$;

QMA is the natural generalization of NP to the quantum setting, but it is very important due to it's tight connections with condensed-matter physics. More concretely, let's consider the following promise problem.

Definition 12 ($k - \mathsf{LH}_{\alpha,\beta}$)**.** *An instance to the k-local Hamiltonian problem is the description of* $H = \frac{1}{m}\sum_{i=1}^{m} H_i$*, where each* H_i *acts on k out of an n qubit system and* $||H_i|| \leq 1$*. Given two threshold* α, β *such that* $\alpha < \beta$*, decide which of the following holds, given the promise that one is true:*

> ***Yes.*** *There exists a quantum state* $|\psi\rangle$ *such that* $\langle\psi| H |\psi\rangle \leq \alpha$.
>
> ***No.*** *For all quantum states* $|\psi\rangle$ *it holds that* $\langle\psi| H |\psi\rangle \geq \beta$.

We want to understand the hardness of $k - \mathsf{LH}_{\alpha,\beta}$ for different values of k, α, β.

Theorem 7 (Quantum Cook-Levin theorem)**.** *For some* $\beta - \alpha \geq \frac{1}{poly(n)}$, $\log(n) - \mathsf{LH}_{\alpha,\beta}$ *is* QMA*-complete.*

We actually know the QMA-completeness of much more structured Hamiltonians.

In order to prove Theorem 7, we need to show that $\log(n) - \mathsf{LH}_{\alpha,\beta} \in \mathsf{QMA}$, which we will present in Section 5.1, and that $\log(n) - \mathsf{LH}_{\alpha,\beta}$ is QMA-hard, which we briefly describe in Section 5.2 and whose simplified proof is one of the exercises.

5.1 Containment in QMA

To prove that $LH_{a,b}$ is in QMA, our goal is to be able to estimate $\langle\psi| H |\psi\rangle$ given a copy of the state $|\psi\rangle$. For that, we will use the Quantum Phase Estimation (Lecture 1).

Let us consider a k-local term H_i, and its eigendecomposition $\{(|\phi_{i,j}\rangle, \lambda_{i,j}\}$. Given some $|\phi_{i,j}\rangle$ QPE with the operator $U_i = e^{iH_i}$ can find $\tilde{\lambda}_{i,j}$ in time $poly(\frac{1}{\epsilon})$ such that $|\tilde{\lambda}_{i,j} - \lambda_{i,j}| < \epsilon$.

Therefore we can consider the following QMA-verification

1. Pick $i \in [m]$ uniformly at random
2. Apply QPE with the operator $U = e^{iH_i}$ on the witness state $|\psi\rangle$ with precision $\frac{\beta-\alpha}{2}$, and get estimate $\tilde{\lambda}$.
3. Accept if value $\tilde{\lambda} \leq \frac{\alpha+\beta}{2}$, and reject otherwise.

Let us suppose that there exists $|\psi^*\rangle$ such that $\langle\psi^*| H |\psi^*\rangle \leq \alpha$. Then, we have that

$$\begin{aligned}
\mathbb{E}[\tilde{\lambda}] &= \frac{1}{m}\sum_{i\in[m]} \Pr[QPE(H_i, |\psi^*\rangle) \\
&< \frac{1}{m}\sum_{i\in[m]}\sum_{j} \lambda_{i,j}|\langle\psi^* |\phi_{i,j}\rangle\langle\phi_{i,j}| \psi^*\rangle| + \frac{\beta-\alpha}{2} \\
&= \frac{1}{m}\sum_{i\in[m]} |\langle\psi^*| H_i |\psi^*\rangle| + \frac{\beta-\alpha}{2} \\
&\leq \alpha + \frac{\beta-\alpha}{2} \\
&= \frac{\alpha+\beta}{2}.
\end{aligned}$$

Therefore, we have that $\mathbb{E}[\tilde{\lambda}] \leq \frac{\alpha+\beta}{2}$, but we need to show that with high probability, $\tilde{\lambda} \leq \frac{\alpha+\beta}{2}$. On the other hand, suppose that for all $|\psi\rangle$, $\langle\psi| H |\psi\rangle \geq \beta$. Then, we have that

$$\begin{aligned}
\mathbb{E}[\tilde{\lambda}] &= \frac{1}{m}\sum_{i\in[m]} \Pr[QPE(H_i, |\psi^*\rangle) \\
&> \frac{1}{m}\sum_{i\in[m]}\sum_{j} \lambda_{i,j}|\langle\psi^* |\phi_{i,j}\rangle\langle\phi_{i,j}| \psi\rangle| - \frac{\beta-\alpha}{2} \\
&= \frac{1}{m}\sum_{i\in[m]} |\langle\psi| H_i |\psi\rangle| - \frac{\beta-\alpha}{2} \\
&\geq \beta - \frac{\beta-\alpha}{2} \\
&\geq \frac{\alpha+\beta}{2}.
\end{aligned}$$

5.2 QMA hardness

The idea of the proof of the QMA-hardness of $\log -$LH is very similar to the equivalence between the circuit model and adiabatic computation.

We define a local Hamiltonian instance H_C that comes from the circuit verification $C = U_T...U_1$. Remember that this circuit comes from a QMA verification, and we assume that if x is a yes-instance, then there exists some $|\psi\rangle$ such that $||(|1\rangle\langle 1| \otimes I)C|x\rangle|\psi\rangle|0\rangle^{\otimes q(|x|)}||^2 \geq 1 - 2^{-n}$; and if x is a no-instance, then for all $|\psi\rangle$, we have that$||(|1\rangle\langle 1| \otimes I)C|x\rangle|\psi\rangle|0\rangle^{\otimes q(|x|)}||^2 \leq 2^{-n}$.

Let $m = T + n + q + 2$. We define

$$H_C = \frac{1}{m}\left(H^{out} + \left(\sum_{j\in[q]} H_j^{init}\right) + \left(\sum_{j\in[n]} H_j^{input}\right) + \left(\sum_{t\in[T]} H_t^{prop}\right) + H_{T+1}^{prop}\right).$$

initialization For $j \in [q]$, $H_j^{init} = |0\rangle\langle 0|_C \otimes |1\rangle\langle 1|_{A_j}$, and for $j \in [n]$, $H_j^{input} = |0\rangle\langle 0|_C \otimes |\overline{x_j}\rangle\langle\overline{x_j}|_{I_j}$,

propagation Let G_t be the set of qubits on which U_t acts non-trivially.
$H_0^{prop} = \frac{1}{2}\Big(|0\rangle\langle 0|_C \otimes I_{G_1} + |1\rangle\langle 1| \otimes I - |1\rangle\langle 0| \otimes U_1 - |0\rangle\langle 1| \otimes U_1^{\dagger}\Big)$
$H_T^{prop} = \frac{1}{2}\Big(|T-1\rangle\langle T-1|_C \otimes I_{G_{T-1}} + |T\rangle\langle T| \otimes I - |T\rangle\langle T-1| \otimes U_{T-1} - |T-1\rangle\langle T| \otimes U_T^{\dagger}\Big)$
For $1 \leq t \leq T-1$,
$H_t^{prop} = \frac{1}{2}\Big(|t-1\rangle\langle t-1|_C \otimes I_{G_t} + |t\rangle\langle t| \otimes I - |t\rangle\langle t-1| \otimes U_t - |t-1\rangle\langle t| \otimes U_t^{\dagger}\Big)$.

output $H^{out} = |T\rangle\langle T|_C \otimes |0\rangle\langle 0|_O$.

In this definition, we have that the register C corresponds to the clock register (as seen in class), the register A_j corresponds to the j-th ancilla qubit; the register I_j corresponds to the j-th bit of the instance; the register G_i corresponds to the qubits touched by the gate i.

As we will see in the exercise, this construction forces low-energy states to be of the form $|\mathsf{hist}\rangle = \frac{1}{\sqrt{T+1}}\sum_{t=\{0,\ldots,T\}} |t\rangle \otimes U_t \ldots U_1(|x\rangle\,|\psi\rangle\,|0\rangle^{\otimes q})$, and if this is the case, then the H^{out} term will penalize no-instances, giving it more energy.

References

[1] Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh V. Vazirani. Strengths and weaknesses of quantum computing. *SIAM J. Comput.*, 26(5):1510–1523, 1997.

[2] Ran Raz and Avishay Tal. Oracle separation of BQP and PH. *J. ACM*, 69(4):30:1–30:21, 2022.

[3] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Rev.*, 41(2):303–332, 1999.

A Chernoff bound

Theorem 8 (Chernoff bound)**.** *Let $X_1, \ldots, X_m \in \{0,1\}$ be random variables such that for each $i = 1, \ldots, m$ we have that*

$$X_i = \begin{cases} 1, & \textit{with probability } p \\ 0, & \textit{with probability } 1-p \end{cases}.$$

Let also $X = \sum_{i=1}^{m} X_i$ and $\mu = \mathbb{E}[X] = pm$. We have that for any $0 \leq \delta \leq 1$,

$$Pr\left(|X - \mu| \geq \mu\delta\right) \leq e^{-\delta^2\mu/3}.$$