Advanced Quantum Information and Computing

Exercise Sheet 4

Exercise 1 (Birthday paradox: an easy proof)**.** *Our goal in this exercise is to prove a weaker form of the birthday paradox (which has the advantage to benefit from an elementary proof while giving the "good" result). Suppose that we build two lists $\mathcal{L}_1, \mathcal{L}_2$ with L elements picked uniformly at random and independently among a set of size N. By definition,*

$$\mathcal{L}_1 = (\mathbf{X}_1, \ldots, \mathbf{X}_L) \quad \textit{and} \quad \mathcal{L}_2 = (\mathbf{Y}_1, \ldots, \mathbf{Y}_L)$$

where the $\mathbf{X}_i$'s and $\mathbf{Y}_i$'s are uniformly and independently distributed. Show that

$$\mathbb{E}\left(\sharp\mathcal{L}_1 \cap \mathcal{L}_2\right) = O\left(\frac{L^2}{N}\right)$$

What do you deduce?

Exercise 2 (Eigenvalues of a bipartite graph)**.** *Let (V, E) be a d-regular graph which is bipartite so V can be partitioned into disjoint sets V_1 and V_2, and*

$$\{v, w\} \in E \Longrightarrow (v \in V_1 \text{ and } w \in B) \text{ or } (v \in V_1 \text{ and } w \in V_2)$$

Let $\mathbf{P}$ be the transition matrix of the random walk induced by (V, E). Show that $\mathbf{P}$ has a 1 eigenvalue and also a -1 eigenvalue. Give the corresponding eigenvectors. What can you say about performing a quantum walk on such a graph?

Exercise 3 (Finding a triangle via Grover's algorithm)**.** *We consider a graph (V, E). Let $n \stackrel{def}{=} \sharp V$ and $m \stackrel{def}{=} \sharp E$. The graph is undirected so $\{i, j\} \in E \Leftrightarrow \{j, i\} \in E$ and without self-loops so $\{i\} \notin E$ for each $i \in V$. We have access to an efficient classical circuit that computes the following function*

$$f_E(i, j) = \begin{cases} 1 \text{ if } \{i, j\} \in E \\ 0 \text{ otherwise} \end{cases}$$

A triangle is a triplet (i, j, k) such that $\{i, j\}, \{j, k\}, \{i, k\} \in E$.

1. *Use Grover's algorithm to find a quantum algorithm that finds a triangle in time $O(n^{3/2})$ if a triangle exists.*
2. *Find a quantum algorithm that finds an edge in time $O\left(\sqrt{\frac{n^2}{m}}\right)$. Argue that the edge found is a random edge from the set of all edges.*
3. *Given an edge $\{i,j\}$, find an algorithm that determines whether there exists k such that (i,j,k) is a triangle in time $O(\sqrt{n})$.*
4. *From there, constructs an algorithm that finds a triangle (i,j,k) (if it exists) in time $O(\sqrt{\frac{n^2}{m}}+\sqrt{n})$ and that succeeds with probability at least $\frac{1}{m}$.*
5. *Use the amplitude amplification technique to design a quantum algorithm for finding a triangle in time $O(n+\sqrt{mn})$.*
6. *Compare this complexity with the one from Question 1. When is it better? Can it be worse?*

Exercise 4 (Finding a triangle via a first quantum random walk)**.** *We studied in the previous algorithm Grover and amplitude amplification approaches to solve the triangle problem (we will use the same notation). We now show how quantum walks can improve the best algorithms for triangle finding. We assume that it exists a single triangle. Furthermore, we also have access to the following oracle*

$$\mathbf{O}_E \left|u\right\rangle \left|v\right\rangle \left|b\right\rangle \rightarrow \left|u\right\rangle \left|v\right\rangle \left|b \oplus \text{“}\{u,v\} \in E''\right\rangle .$$

where “$\{u,v\} \in E''$ is a bit which is equal to 1 if $\{u,v\} \in E$ and 0 otherwise. We consider as a cost measure only the query complexity, i.e., the number of calls to $\mathbf{O}_E$.

For a parameter r, we construct a graph $H = (V_H, E_H)$ on which we will perform a quantum walk.

- *Each $S \in V_H$ is of the form $S(\text{ver}), S(\text{edges})$ where $S(\text{ver})$ contains a list of distinct vertices $v_1, \ldots, v_r \in V$. $S(\text{edges})$ contains all the values $e_{i,j} = 1$ if $\{v_i, v_j\} \in E$ and $e_{i,j} = 0$ otherwise, for each $i, j \in \{1, \ldots, r\}$ and $i < j$.*
- *A pair $(S_1, S_2) \in E_H$ if and only if you can construct $S_2(\text{ver})$ from $S_1(\text{ver})$ by removing exactly one element from $S_1(\text{ver})$ and adding another element. More formally:*

$$(S_1, S_2) \in E_H \Longleftrightarrow \exists v_1 \in S_1(\text{ver}), \exists v_2 \in S_2(\text{ver}) \backslash S_1(\text{ver}),$$

such that,

$$S_2(\text{ver}) = (S_1(\text{ver})\backslash\{v_1\}) \cup \{v_2\}.$$

- *An vertex $S \in V_H$ is marked if there exist $v_i, v_j \in S(\text{ver})$ and $u \in V$ such that v_i, v_j, u is a triangle of G.*

We want to perform a quantum walk on this graph H, which is a Johnson graph $J(n, r)$.

1. *What is the fraction ε of marked vertices and what is the spectral gap δ of H?*
2. *Show that the setup cost is $O(r^2)$ and the update cost is $O(r)$.*
3. *Find a quantum algorithm that checks whether a vertex is marked in time $O(r\sqrt{n})$, what is then the total running time of the quantum walk to find a triangle in G? Is it better than Grover's approach?*

Exercise 5 (A better quantum walk to find a triangle)**.** *The goal of this exercise is to find a better quantum algorithm by improving the checking procedure. This algorithm will use another quantum walk. We fix a vertex $S = (S(\text{ver}), S(\text{edges})) \in V_H$ and we want to check whether S is marked or not. We also fix a vertex $u \in V$, and construct the graph $H' = (V', E')$ as follows:*

- *Each $S' \in V'$ is of the form $(S'(\text{ver}), S'(\text{edges}), S'(\text{edges}_u))$ where*

$$S'(\text{ver}) \subset S(\text{ver}) \text{ with } \sharp S'(\text{ver}) = r^{2/3}.$$

We therefore write $S'(\text{ver}) = v_1, \ldots, v_{r^{2/3}}$. $S'(\text{edges})$ contains all the strings $e_{ij} = 1$ if $\{v_i, v_j\} \in E$ and $e_{ij} = 0$ otherwise for each $v_i, v_j \in S'(\text{ver})$. $S'(\text{edges}_u)$ contains all the strings $e_i^u = 1$ if and only if $\{v_i, u\} \in E$ and $e_i^u = 0$ otherwise.

- *$(S_1', S_2') \in E'$ if and only you can go construct $S_2'(\text{ver})$ from $S_1'(\text{ver})$ by removing exactly one element from $S_1'(\text{ver})$ and adding another element. More formally:*

$$(S_1', S_2') \in E \iff \exists v_1 \in S_1'(\text{ver}), \exists v_2 \in S_2'(\text{ver})\backslash S_1'(\text{ver})$$

such that

$$S_2'(ver) = (S_1'(ver)\backslash\{v_1\}) \cup \{v_2\}.$$

- *An element $S' \in V'$ is marked if and only if $\exists v_i, v_j \in S'(ver)$ such that (v_i, v_j, u) is a triangle in G.*

We want to perform a quantum walk on the graph H', which is a Johnson graph $J(r, r^{2/3})$. Recall that we consider as a cost measure only the query complexity, i.e., the number of calls to $\mathbf{O}_E$.

1. *What is the setup cost of this random walk? Recall that the edges in $S'(\text{edges})$ have already been queried (they are in $S(\text{ver})$) and can be computed at 0 cost.*

2. *Show that the update and the checking cost are respectively $O(1)$ and 0.*

3. *Assume there is a pair $v_i, v_j \in S$ such that v_i, v_j, u form a triangle in G. In this case, what is the fraction ε of marked vertices in H'. In this case, show that the quantum walk can find a vertex in time $O(r^{2/3})$.*

4. *From there, show that the checking cost from Exercise 1 can be reduced to $O(r^{2/3}\sqrt{n})$. What is therefore the total running time of the quantum walk described in the previous exercise?*

Exercise 6 (Checking matrix multiplication)**.** *We consider three matrices $\mathbf{A}, \mathbf{B}, \mathbf{C} \in \{0,1\}^{n\times n}$, to which we have the following query access*

$$\mathbf{O}_{\mathbf{M}} : |i\rangle\,|j\rangle\,|b\rangle \to |i\rangle\,|j\rangle\,|b \oplus \mathbf{M}_{i,j}\rangle\,.$$

for the three matrices $\mathbf{M} = \mathbf{A}, \mathbf{B}, \mathbf{C}$. Our goal is to check whether $\mathbf{AB} = \mathbf{C}$ or not.

For sets $\mathcal{S}, \mathcal{T} \subseteq \{1, \dots, n\}$, we consider the matrices $\mathbf{A}_{\mathcal{S}}$ and $\mathbf{B}_{\mathcal{T}}$ where $\mathbf{A}_{\mathcal{S}} \in \{0,1\}^{\sharp\mathcal{S}\times n}$ is the sub-matrix with lines in $\mathcal{S}$, and $\mathbf{B}_{\mathcal{T}} \in \{0,1\}^{n\times\sharp\mathcal{T}}$ is the sub-matrix of $\mathbf{B}$ that only considers columns in $\mathcal{T}$. The multiplication $\mathbf{A}_{\mathcal{S}} \cdot \mathbf{B}_{\mathcal{T}}$ outputs a matrix in $\{0,1\}^{\sharp\mathcal{S}\times\sharp\mathcal{T}}$. Let $\mathbf{C}_{\mathcal{S},\mathcal{T}}$ be the sub-matrix of $\mathbf{C}$ that consists of lines in $\mathcal{S}$ and columns in $\mathcal{T}$. If $\mathbf{AB} = \mathbf{C}$ then $\mathbf{A}_{\mathcal{S}} \cdot \mathbf{B}_{\mathcal{T}} = \mathbf{C}_{\mathcal{S},\mathcal{T}}$.

Construct a quantum walk on a graph (V, E). In each vertex $v_{\mathcal{S},\mathcal{T}} \in V$, put all the bits of $\mathbf{A}_{\mathcal{S}}, \mathbf{B}_{\mathcal{T}}, \mathbf{C}_{\mathcal{S},\mathcal{T}}$ for subsets $\mathcal{S}, \mathcal{T}$ such that $\sharp\mathcal{S} = \sharp\mathcal{T} = r$. A vertex is marked if $\mathbf{A}_{\mathcal{S}} \cdot \mathbf{B}_{\mathcal{T}} \neq \mathbf{C}_{\mathcal{S},\mathcal{T}}$. You put an edge between two vertices $v_{\mathcal{S},\mathcal{T}}$ and $v_{\mathcal{S}',\mathcal{T}'}$ is $\mathcal{S} = \mathcal{S}'$ and $\mathcal{T}, \mathcal{T}'$ differ by at most 1 elements or $\mathcal{T} = \mathcal{T}''$ and $\mathcal{S}, \mathcal{S}'$ differ by at most one element. The graph you construct will have spectral gap

$$\delta = O\left(\frac{1}{r}\right).$$

Show that the running time (in the number of queries) of this quantum walk is $O(n^{5/3})$, for a well chosen parameter r.Give the running time of the different steps: Setup, Update, Check and the fraction of marked vertices.

Exercise 7 (About 3-SAT)**.** *A* $3-SAT$ *instance* Φ *over* n *Boolean variables* $x_1, \ldots, x_n$ *is a formula which is the* **AND** *of a number of clauses, each of which is an* **OR** *of* 3 *variables or their negations. For example,*

$$\Phi(x_1, \ldots, x_4) = (x_1 \textbf{ OR } x_2 \textbf{ OR } x_3) \textbf{ AND } (x_2 \textbf{ OR } x_3 \textbf{ OR } x_4)$$

is a 3*-SAT formula with* 2 *clauses. A satisfying assignment is a setting of the* n *variables such that* $\Phi(x_1, \ldots, x_n) = 1$ *(i.e, TRUE). You may assume the number of clauses is at most some polynomial in* n*. In general it is* NP*-hard to find a satisfying assignment to such a formula. Brute force would try out all* 2^n *possible truth assignments, but something much better is possible: consider the following simple algorithm of Schöning, which is a classical random walk on the set of all* $N = 2^n$ *truth assignments:*

(*a*) *Start with a uniformly random* $\mathbf{x} \in \{0,1\}^n$

(*b*) *Repeat the following at most* $3n$ *times: if* $\Phi(\mathbf{x}) = 1$ *then STOP, else find the leftmost clause that is false, randomly choose one of its* 3 *variables and flip its value.*

One can show that this algorithm has probability at least $(3/4)^n/\sqrt{5n}$ *of finding a satisfying assignment (if* Φ *is satisfiable). You may assume this without proof.*

1. *Use the above to give a classical algorithm that finds a satisfying assignment with high probability in time* $(4/3)^n \cdot p(n)$*, where* $p(n)$ *is some polynomial factor.*

2. *Give a quantum algorithm that finds a satisfying assignment (with high probability) in time* $\sqrt{(4/3)^n} \cdot p(n)$*.*