Quantum Computing 2/24/21

Lecture 10: Amplitude Amplification

Instructor: Dieter van Melkebeek Scribe: Roger Waleffe

In this lecture, we generalize the principles of the previous lecture on quantum search and present amplitude amplification—a basic paradigm in quantum algorithms. We start by defining the problem statement and then present the algorithm. We then discuss how to eliminate error and how to relax the requirement that the total weight of "good" solutions is known ahead of time.

1 Amplitude Amplification

1.1 Problem Statement

The problem setup for amplitude amplification is similar to that of Grover's search algorithm discussed previously. In particular, we are given:

- a black-box Boolean function $f : \{0,1\}^n \to \{0,1\}$ which maps "good" inputs x to 1 and "bad" inputs to 0,
- a unitary circuit A operating on n qubits such that $A\left|0^n\right\rangle = \sum_x \alpha_x \left|x\right\rangle$ produces nonzero amplitude α_x on at least one x with $f(x) = 1$,
- and the total weight $p \doteq \sum_{x:f(x)=1} |\alpha_x|^2$.

The second assumption implies that there exists at least one "good" x and that $p > 0$. In Section 3 we relax the requirement that the total weight of "good" inputs p is known ahead of time.

The state $A\left|0^n\right\rangle = \sum_x \alpha_x \left|x\right\rangle$ can be written as a superposition of the "good" states $\left|G\right\rangle = \frac{1}{\sqrt{p}} \sum_{x:f(x)=1} \alpha_x \left|x\right\rangle$ and "bad" states $\left|B\right\rangle = \frac{1}{\sqrt{1-p}} \sum_{x:f(x)=0} \alpha_x \left|x\right\rangle$:

$$A\left|0^n\right\rangle = \sqrt{1-p}\left|B\right\rangle + \sqrt{p}\left|G\right\rangle. \tag{1}$$

The goal of amplitude amplification is to output $\left|G\right\rangle$ in Equation 1 with probability greater than or equal to 1/2 along with a success indicator (a bit which is 1 if $\left|G\right\rangle$ is the output and 0 otherwise). As was the case for quantum search, the high level idea is to amplify the weights of $\left|G\right\rangle$ while reducing the weights of $\left|B\right\rangle$.

Remark 1. The problem statement for amplitude amplification differs from that of quantum search in subtle yet distinct ways. In particular, for quantum search, the goal was to identify *a single* x such that $f(x) = 1$ by amplifying the weights α_x for "good" inputs x and then performing a measurement. Thus, the output of quantum search *was classical.* In this lecture, however, the output is instead the superposition of "good" states $\left|G\right\rangle$ itself. This means that amplitude amplification is actually a *quantum subroutine* which can be used as a building block for larger quantum algorithms.

A second consequence of the difference in outputs is that for amplitude amplification we should consider *all* coefficients in the state $\left|G\right\rangle$, not just that *one* "good" state is sufficiently amplified to be measured with high probability. In particular, amplitude amplification should proceed without affecting the relative weight between good inputs—the ratio of the amplitudes α_x and $\alpha_{x'}$ of good states should not change; only the ratio of the amplitudes of good versus bad states can change. While the algorithm we presented for quantum search satisfies this requirement, it may be less important for quantum search in general, so long as at least one "good" x ends up with a high probability of measurement.

Finally, note that quantum search as discussed in the previous lecture can simply be viewed as a special case of amplitude amplification with a measurement at the end. The specific case of $A = H^{\otimes n}$ (the n-fold Hadamard) is also noteworthy as it produces a uniform superposition over all x.

1.2 Algorithm

The algorithm for amplitude amplification proceeds in a similar fashion to the search algorithm discussed last lecture. The main difference comes when it is time to extract the output. As described in Algorithm 1, we start from the state $A\,|0^n\rangle$ then apply the following k times: we first apply a phase flip on all "good" x using R_{bad} (which requires using the quantum version of the black-box function) and then reflect around the start state using $R_{initial}$. See the previous lecture for more details.

```
Algorithm 1: Amplitude Amplification.
Start with $A\,|0^n\rangle$
for $k$ times:
    Apply $R_{bad}$ (using $U_f$): Phase flip on all $|x\rangle$ with $f(x) = 1$
    Apply $R_{initial} \doteq AR_{|0^n\rangle}A^{-1}$: Reflection around start state
```

1.2.1 Two-dimensional State Analysis

The iterative procedure involved in amplitude amplification can be analyzed geometrically in two-dimensional space as done in the previous lecture. We summarize the idea here for completeness. First we express the state of the system after k iterations as $|\psi_k\rangle = \beta_k\,|B\rangle + \gamma_k\,|G\rangle = \cos\theta_k\,|B\rangle + \sin\theta_k\,|G\rangle$. As such, the system corresponds to a point on the unit circle as shown in Figure 1. Initially we have that $\beta_0 = \sqrt{1-p} = \cos\theta_0$ and $\gamma_0 = \sqrt{p} = \sin\theta_0$. Then each iteration applies R_{bad} followed by $R_{initial}$. Geometrically, R_{bad} consists of a reflection about the $|B\rangle$ axis (as it flips the phase of all "good" states x). In the two-dimensional $|B\rangle$-$|G\rangle$ plane, $R_{initial}$ corresponds to a reflection about $A\,|0^n\rangle$. The combined result of the two operations is a counterclockwise rotation over $2\theta_0$ for each iteration as shown in Figure 1. More detailed diagrams for each component of an iteration can be found in the previous lecture.

1.2.2 Extracting $|G\rangle$

Recall that the goal of amplitude amplification is to output $|G\rangle$. After k iterations of the algorithm, we can do so as follows: We introduce a fresh ancilla qubit initially set to $|0\rangle$ creating the state $\sum_x \alpha_x^{(k)}\,|x\rangle\,|0\rangle$, with $\alpha_x^{(k)}$ the final amplitude for state x after the k-step amplification process. We then apply the black box function U_f to the quantum state resulting in the state $\sum_x \alpha_x^{(k)}\,|x\rangle\,|f(x)\rangle$ (recall that $U_f\,|x\rangle\,|y\rangle = |x\rangle\,|y \oplus f(x)\rangle$). The final step is to measure the ancilla qubit. If we observe $|1\rangle$ then the remaining quantum state collapses to those which are consistent

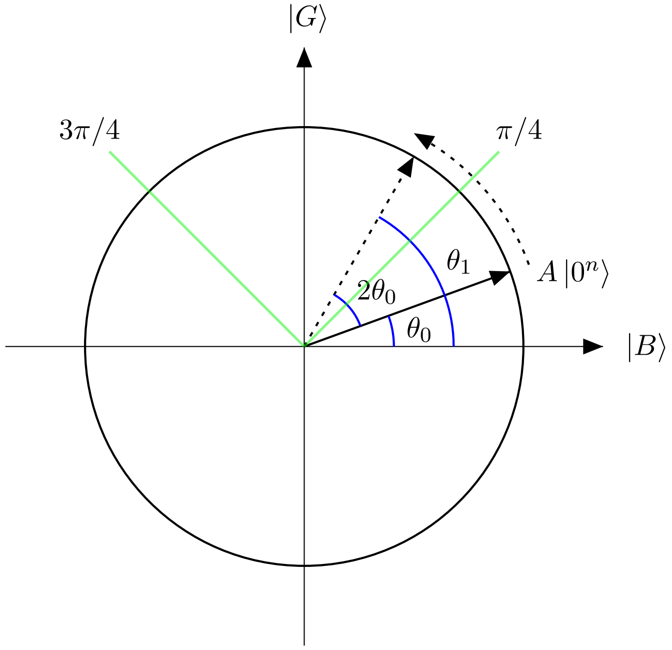

Figure 1: Depiction of amplitude amplification interation process in two-dimensional plane. The initial state starts with angle θ_0. Each iteration rotates the state in the $|B\rangle$-$|G\rangle$ plane by $2\theta_0$. If the angle after k iterations θ_k is between $[\pi/4, 3\pi/4]$ then the probability of extracting $|G\rangle$ is $\geq 1/2$.

with the measurement—i.e. states that satisfy $f(x) = 1$. This is exactly the state $|G\rangle$ (up to a normalization). Likewise if we measure $|0\rangle$ for the ancilla qubit then the remaining state is $|B\rangle$. *Notice that the ancilla qubit is exactly the success indicator desired in our problem statement.*

1.2.3 Number of Iterations and Error Bound

In order to ensure that the probability of outputting $|G\rangle$ is greater than or equal to 1/2 we must ensure that after k iterations $\gamma_k^2 \geq \beta_k^2$. In other words, in the two-dimensional state view, we wish for the final angle θ_k after k iterations to end up as close to plus or minus $\pi/2$ as possible (Figure 1). This can be done by choosing k as described in the previous lecture. Ideally $\theta_k = (2k+1)\theta_0 = \frac{\pi}{2}$. This implies that k should be set to $k^* \doteq \frac{1}{2}(\frac{\pi}{2\theta_0} - 1)$. In general, k^* is non-integral (in the case that it is an integer, then $|G\rangle$ can be extracted with probability 1). We take the number of iterations k to be $\lfloor k^* \rceil$, the closest integer to k^*. This ensures that $\theta_k \in [\frac{\pi}{4}, \frac{3\pi}{4}]$ and thus that the probability of extracting $|G\rangle$ rather than $|B\rangle$ is greater than or equal to 1/2. In other words, $\Pr[\text{success}] \geq \frac{1}{2}$ as desired. Recall also that setting k in this manner results in $O(\frac{1}{\sqrt{p}})$ iterations (and thus black-box queries) because $k^* = \Theta(1/\theta_o) = \Theta(1/\sqrt{p})$ since $\sin\theta_0 = \sqrt{p}$ and $\sin x \sim x$ for small x).

2 Error Elimination

In this section we discuss how to eliminate the error in amplitude amplification—i.e., how to ensure we can produce $|G\rangle$ with certainty rather than just with probability greater than or equal to 1/2.

This constitutes the solution to the exercise posed in the previous lecture.

2.1 Key Idea

The problem with amplitude amplification as described above is that it is often impossible to rotate the initial angle θ_0 to $\pi/2$ using an integral number k of angle $2\theta_0$ rotations. Thus we are unable to produce a final state $|G\rangle$, and are instead only able to produce a superposition $|\psi_k\rangle = \beta_k |B\rangle + \gamma_k |G\rangle$ with $\gamma_k^2 \geq \beta_k^2$. To alleviate this issue, the key idea is to lower θ_0 a little bit to $\tilde{\theta}_0$ such that it is possible to end up exactly at $\pi/2$ after a integer sequence of iterations. To reduce the angle we will make use of an additional ancilla qubit.

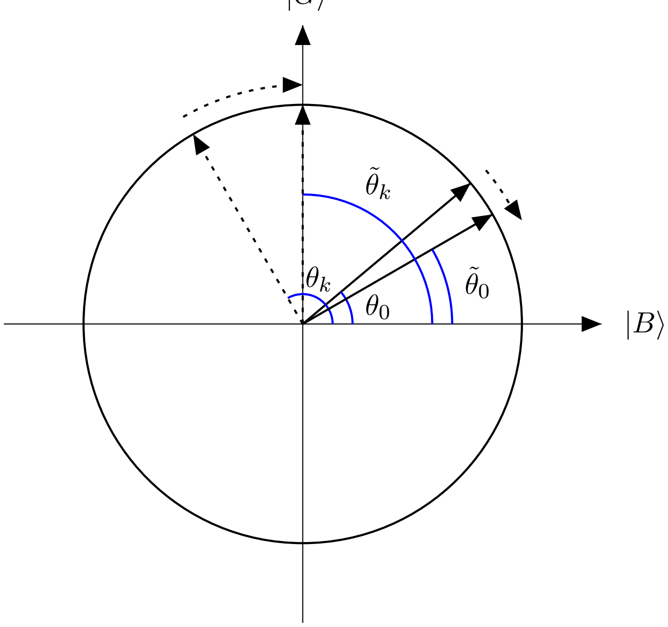

Figure 2: Depiction of the key idea in amplitude amplification error elimination: By reducing θ_0 to $\tilde{\theta}_0$ we can move $\theta_k > \pi/2$ to $\tilde{\theta}_k = \pi/2$ and ensure that $|G\rangle$ is extracted with certainty.

2.2 Solution

Recall that we defined k^* such that after k^* iterations amplitude amplification produces the state $|G\rangle$. In general k^* is non-integral. Consider $k = \lceil k^* \rceil$. Then we know that after k iterations $\theta_k \geq \pi/2$. In other words, we may overshoot $|G\rangle$ in the two-dimensional $|B\rangle$-$|G\rangle$ plane. Following the key idea above, we aim to lower $\theta_0 \to \tilde{\theta}_0$ such that after k iterations, instead of overshooting, we get exactly $\theta_k = \pi/2$. Lowering $\theta_0 \to \tilde{\theta}_0$ means that $\tilde{\theta}_0 \leq \theta_0$ and implies that $\tilde{p} \doteq \sin^2(\tilde{\theta}_0) \leq \sin^2(\theta_0) = p$. Concretely, we wish to choose $\tilde{\theta}_0$ such that $(2k+1)\tilde{\theta}_0 = \frac{\pi}{2}$. The geometric interpretation is shown in Figure 2.

To implement this reduction in angle from $\theta_0 \to \tilde{\theta}_0$ we apply amplitude amplification on a system expanded with an additional ancilla qubit: $\tilde{A}\,|0^{n+1}\rangle = A\,|0^n\rangle\, U\,|0\rangle$ where U is some unitary matrix such that $U\,|0\rangle \doteq \alpha_0\,|0\rangle + \alpha_1\,|1\rangle$. After expanding the size of the system from n to $n+1$

dimensions, we must also "expand" the function $f(x)$ to operate on $n+1$ dimensional vectors rather than n dimensional vectors. We define $\tilde{f}(xb) = f(x) \cdot b$, where b is the additional ancilla bit. "Good" inputs are now those which have $\tilde{f}(xb) = 1$. These are the inputs for which $f(x) = 1$ and $b = 1$. This means that our new superposition of "good" inputs x is now $|\tilde{G}\rangle = |G\rangle |1\rangle$, and that the weight of $|\tilde{G}\rangle$ is $\tilde{p} = p \cdot |\alpha_1|^2$. The only remaining question is then: can we find a unitary matrix U such that $U|0\rangle \doteq \alpha_0 |0\rangle + \alpha_1 |1\rangle$ with $|\alpha_1|^2 = \tilde{p}/p$? The answer is yes. We use the matrix

$$U = \begin{bmatrix} \alpha_0 & \alpha_1 \\ \alpha_1 & -\alpha_0 \end{bmatrix} \tag{2}$$

with $\alpha_1 = \sqrt{\tilde{p}/p}$ and $\alpha_0 = \sqrt{1 - \tilde{p}/p}$. Since $\tilde{p} \leq p$, $\alpha_1 \leq 1$ and we can chose $\alpha_0 \in \mathbb{R}$ such that the first column of U has two norm one. The second column is chosen to be orthogonal to the first without consequence, as we are only interested in the application of U to $|0\rangle$.

Remark 2. Note that if all inputs x are "good", then amplitude amplification requires zero iterations. Amplitude amplification only requires one iteration if $p \geq 1/4$. In this case p will be reduce to $\tilde{p} = 1/4$ meaning $\tilde{\theta}_0 = \pi/6$. With one iteration, $\pi/6$ will then rotate to $3\pi/6 = \pi/2$.

3 Amplitude Amplification with Unknown Initial Weight

We now discuss how to perform amplitude amplification when the total weight of "good" inputs $p = \sum_{x:f(x)=1} |\alpha_x|^2$ is unknown (p is still assumed to be larger than 0). The problem setup remains the same as described in Section 1.1 except for the aforementioned change. Additionally, rather than focus on the number of black-box function calls required to output $|G\rangle$ with probability $\geq 1/2$, we now focus on the expected number of black-box function queries required to output $|G\rangle$. The two quantities differ by at most a factor of two since the expected number of trials required for success in a Bernoulli experiment with probability $1/2$ is two.

We will develop the algorithm for amplitude amplification with unknown weight p in a series of stages which tweak the algorithm presented above. Later in the course we will see how this new algorithm emerges more naturally. Note that we can still use p in the analysis of the algorithm. We can not however use it in the algorithm, e.g., to determine the number of iterations k^* as done previously.

3.1 First Attempt

Since p is unknown, we are unable to compute the number of iterations k^* as described in Section 1.2.3 (unknown p means unknown θ_0). A natural first attempt then, consists of trying $k = 0, 1, 2, 3 \ldots$ iterations until there is a success (recall we have a success bit when extracting $|G\rangle$). For each value of k we must first reset the system to $A|0^n\rangle$ and then perform the sequence of iterations (see below remark).

To count the number of expected black-box queries required for this algorithm attempt, we can split the trial values of k into those which are less than and those which are greater than k^* (even though k^* is unknown, it can still be used for analysis). The number of queries for each trial is $k+1$ (k for the iterations, 1 for the ancilla qubit introduced to extract $|G\rangle$). Thus, the number of queries for the trials of all values of k up to k^* is:

$$\sum_{k=0}^{k^*} (k+1) = \Theta((k^*)^2) = \Theta((1/\sqrt{p})^2) = \Theta(1/p). \tag{3}$$

The first equality follows from evaluating the sum, while the second results from the fact that $k^* = \Theta(1/\theta_o) = \Theta(1/\sqrt{p})$ as mentioned in Section 1.2.3.

While the algorithm may succeed in outputting $|G\rangle$ before reaching k^*, we assume that the above equation gives the query complexity (i.e., that k often reaches at least k^*, see the below exercise). Notice that this attempt has lost the square root quantum speed up even before k exceeds k^*—the query complexity is $\Theta(1/p)$ rather than $\Theta(1/\sqrt{p})$. The former is the same as the classical setting. In this case we have gained no performance improvement by using amplitude amplification: If the initial state has a probability p of being in $|G\rangle$ then we can simply try to extract $|G\rangle$ and we will succeed with probability p. Viewing this as a Bernoulli experiment, we can expect to succeed after $1/p$ trials.

Remark 3. Retrying for different values of k, and more generally restarting in quantum algorithms, can be non-trivial. In fact, the need to restart can force algorithms to begin with special quantum states rather than general superpositions. For example, why does amplitude amplification start from $A|0^n\rangle$? One reason is that A is needed for reflection in $R_{initial}$. A second reason however, is that it allows for the algorithm to be restarted. In general, algorithms can't go back to their initial quantum superpositions after a measurement has been made, but by starting from a base state $|0^n\rangle$ rewinding is possible for amplitude amplification.

Exercise 1. *For theory students: Show that Pr[k^* is reached] $= \Omega(1)$. This means that the probability we must try values of k up to k^* is a constant value and thus that the above query complexity is well justified.*

3.2 Second Attempt

The problem with the above attempt which leads to $\Theta(1/p)$ query complexity is that the algorithm spends significant effort trying small values of $k < k^*$ which result in small probabilities of success (extracting $|G\rangle$). A simple approach to mitigate this issue is to use a geometric sequence of k values rather than an arithmetic sequence. In particular, we can try doubling k each time—$k = \lfloor 2^i \rfloor$ with $i = -1, 0, 1, 2, ...$—until we successfully extract $|G\rangle$. Additionally, by only doubling k each time, we can actually ensure that we will not overshoot the region of the $|B\rangle$-$|G\rangle$ plane which measures $|G\rangle$ with probability $\geq 1/2$. In other words, some value of k will have $\theta_k \in [\frac{\pi}{4}, \frac{3\pi}{4}]$. To see why this is true, recall from Section 1.2.3 that $\theta_k = (2k+1)\theta_0$. Thus if we double k to $2k$, $\theta_{2k} \leq 2\theta_k$. This implies that any $\theta_k < \pi/4$ will not result in a $\theta_{2k} > 3\pi/4$, and thus the high probability region $[\frac{\pi}{4}, \frac{3\pi}{4}]$ is not skipped over.

We can analyze the number of queries to the black-box function for this attempt as done above. The number of queries for values of k up to k^* is:

$$\sum_{i=-1}^{\log(k^*)} (\lfloor 2^i \rfloor + 1) = O(k^* \cdot \sum_{i \geq 0} \frac{1}{\lfloor 2^i \rfloor}) = O(k^*) = O(1/\sqrt{p}). \tag{4}$$

While this attempt results in the desired query complexity for $k < k^*$, a key problem remains. If k^* is exceeded the success probability of extracting $|G\rangle$ for $k > k^*$ can be very low (recall we can not stop at k^* because p is unknown, we can only use it for the analysis). We provide intuition for this phenomena by example. Assume that $\theta_0 = \pi/3$. Then for $k = 0$ we have a success probability $\geq 1/2$, but in the chance that we fail and get $|B\rangle$ instead of $|G\rangle$, the next trail of the algorithm will use $k = 1$. In this case $\theta_k = \theta_1 = \pi$ resulting in a zero probability of success! The consequence of this problem is that the query complexity for $k > k^*$ is likely to be worse than $O(1/\sqrt{p})$ implying that the complexity of the whole algorithm will not be $O(1/\sqrt{p})$.

3.3 Third Attempt

From the second attempt above, we know that a geometric sequence for trail values of k leads to $O(1/\sqrt{p})$ black-box queries while $k < k^*$. For these successive values of k, the probability of success increases (θ_k approaches the region $[\frac{\pi}{4}, \frac{3\pi}{4}]$). However, we also know that if we fail to extract $|G\rangle$ before $k > k^*$ then the probability of success can drop considerably. How can we handle this problem? One solution is to view the geometric sequence as an *upper bound* on the number of amplitude amplification iterations for each successive trial, rather than the exact number of iterations itself. To that end, in this attempt, we pick the actual number of iterations k for each trial to be uniformly from $\{0, 1, 2, \ldots \lfloor 2^i \rfloor\ \}$ with $i = -1, 0, 1, 2...$ and continue increasing i until the first success. The intuition here is that the probability of picking a bad number of iterations (e.g., such that the final $\theta_k = \pi$) is low. For this attempt, we continue to double the maximum number of iterations (the "bound") between successive trials.

From the above analysis, we know that the total number of black-box queries before $\lfloor 2^i \rfloor$ exceeds k^* is $O(k^*) = O(1/\sqrt{p})$ (the uniform sampling can only lower the number of black-box queries from attempt two, but does not change the overall scaling). Thus, we focus the analysis here on the case where $\lfloor 2^i \rfloor > k^*$ (i.e., we have not yet had success before $\lfloor 2^i \rfloor$ reaches k^*). Asymptotically, once $\lfloor 2^i \rfloor > k^*$ the uniform sampling effectively results in a number of iterations that produce a final θ_k after amplitude amplification between $[0, 2\pi]$. In this case, half of this range—$[\frac{\pi}{4}, \frac{3\pi}{4}]$ and $[\frac{5\pi}{4}, \frac{7\pi}{4}]$ (the so called "good regions")—has a probability of success $\geq 1/2$. Thus, the probability of success for each trial with $\lfloor 2^i \rfloor > k^*$ is asymptotically: $\Pr[k \text{ in good region}] \cdot \Pr[\text{success} \mid \text{in good region}] \geq \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$.

Thus, every trial with $\lfloor 2^i \rfloor > k^*$ is like a Bernoulli experiment with success probability at least some constant (say 1/4 asymptotically). To analyze the expected number of black-box queries after the bound $\lfloor 2^i \rfloor > k^*$, we introduce i' to count the number of trials for which $\lfloor 2^i \rfloor > k^*$. In other words, $i = i^* + i'$ with $2^{i^*} = k^*$. The expected number of black-box queries after the bound $\lfloor 2^i \rfloor > k^*$ is then the probability of requiring a certain number of trials i' (the probability that all previous trials $i' = 0, 1, 2, \ldots$ failed) times the number of queries for that specific value of i'. This is order:

$$\sum_{i' \geq 0} (\frac{3}{4})^{i'} \cdot (2^i) = \sum_{i' \geq 0} (\frac{3}{4})^{i'} \cdot (k^* \cdot 2^{i'}) = k^* \sum_{i' \geq 0} (\frac{3}{2})^{i'}. \tag{5}$$

We would like this complexity to be $O(k^*) = O(1/\sqrt{p})$, in which case the whole algorithm would be $O(1/\sqrt{p})$ (by combining the complexities before and after k^*). Unfortunately, the above geometric series diverges since $3/2 > 1$.

3.4 Final Attempt

Our final attempt for amplitude amplification with unknown initial weight p (unknown k^*) consists of tweaking attempt three to ensure that the series for the number of queries after the bound $\lfloor 2^i \rfloor > k^*$ converges. The idea is to change from doubling the bound in each successive trial to some other multiplicative constant factor. Assume that the bound on the number of iterations increases by a constant factor $\lambda > 1$ (it must be larger than 1 for the bound to increase). I.e., we pick the number of iterations k for each trial to be uniformly from $\{0, 1, 2, \ldots \lfloor \lambda^i \rfloor \ \}$ with $i = -1, 0, 1, 2...$ and continue until the first success. Using a similar analysis as above in attempt three results in the expected number of black-box queries after the bound $\lfloor \lambda^i \rfloor > k^*$ to be given by:

$$\sum_{i' \geq 0} (\frac{3}{4})^{i'} \cdot (k^* \cdot \lambda^{i'}) = k^* \sum_{i' \geq 0} (\frac{3\lambda}{4})^{i'}. \tag{6}$$

This series converges for $3\lambda/4 < 1$. Thus we have that the number of applications of the black-box is $O(k^*) = O(1/\sqrt{p})$ for $1 < \lambda < 3/4$. This final attempt results in an amplitude amplification algorithm with square root speed up even when the total weight p of the "good" inputs x is unknown.