

Алгоритмы Дойча и Дойча–Йожи

М. В. Волков

Уральский федеральный университет
Институт естественных наук и математики
кафедра алгебры и фундаментальной информатики

2020/2021 учебный год

Рассмотрим двухэлементное множество $\mathbb{F} := \{0, 1\}$.

Рассмотрим двухэлементное множество $\mathbb{F} := \{0, 1\}$.

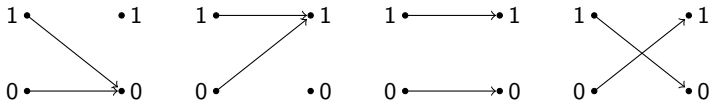
Существуют ровно четыре функции $f: \mathbb{F} \rightarrow \mathbb{F}$.

Две из них — *константы* (функция, которая всегда возвращает 0, и функция, которая всегда возвращает 1), а две нет (тождественная функция и перестановка).

Рассмотрим двухэлементное множество $\mathbb{F} := \{0, 1\}$.

Существуют ровно четыре функции $f: \mathbb{F} \rightarrow \mathbb{F}$.

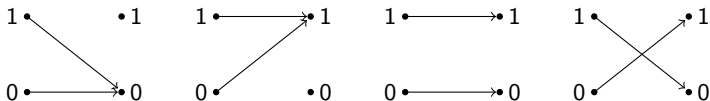
Две из них — *константы* (функция, которая всегда возвращает 0, и функция, которая всегда возвращает 1), а две нет (тождественная функция и перестановка).



Рассмотрим двухэлементное множество $\mathbb{F} := \{0, 1\}$.

Существуют ровно четыре функции $f: \mathbb{F} \rightarrow \mathbb{F}$.

Две из них — *константы* (функция, которая всегда возвращает 0, и функция, которая всегда возвращает 1), а две нет (тождественная функция и перестановка).

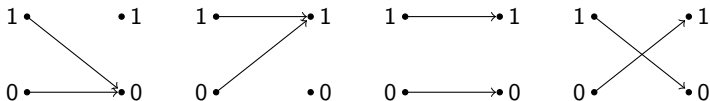


Допустим, что можно лишь **один раз** обратиться к неизвестной функции f как к черному ящику: задать аргумент $x \in \{0, 1\}$ и получить значение $f(x)$.

Рассмотрим двухэлементное множество $\mathbb{F} := \{0, 1\}$.

Существуют ровно четыре функции $f: \mathbb{F} \rightarrow \mathbb{F}$.

Две из них — *константы* (функция, которая всегда возвращает 0, и функция, которая всегда возвращает 1), а две нет (тождественная функция и перестановка).

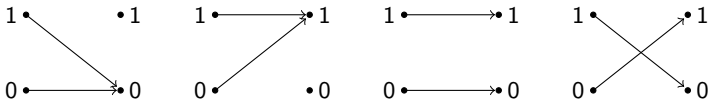


Допустим, что можно лишь **один раз** обратиться к неизвестной функции f как к черному ящику: задать аргумент $x \in \{0, 1\}$ и получить значение $f(x)$. Нужно узнать, является функция f константой.

Рассмотрим двухэлементное множество $\mathbb{F} := \{0, 1\}$.

Существуют ровно четыре функции $f: \mathbb{F} \rightarrow \mathbb{F}$.

Две из них — **константы** (функция, которая всегда возвращает 0, и функция, которая всегда возвращает 1), а две нет (тождественная функция и перестановка).



Допустим, что можно лишь **один раз** обратиться к неизвестной функции f как к черному ящику: задать аргумент $x \in \{0, 1\}$ и получить значение $f(x)$. Нужно узнать, является функция f константой.

Ясно, что никакой классический алгоритм не может решить эту задачу.

Дэвид Дойч в 1985 г. предложил квантовый алгоритм для решения описанной задачи, точнее, ее «квантовой» модификации.

Дэвид Дойч в 1985 г. предложил квантовый алгоритм для решения описанной задачи, точнее, ее «квантовой» модификации.

Хотя алгоритм Дойча решает задачу, не представляющую никакого интереса, мы подробно рассмотрим его, чтобы на простом примере зафиксировать *основные допущения* теории квантовых алгоритмов.

Дэвид Дойч в 1985 г. предложил квантовый алгоритм для решения описанной задачи, точнее, ее «квантовой» модификации.

Хотя алгоритм Дойча решает задачу, не представляющую никакого интереса, мы подробно рассмотрим его, чтобы на простом примере зафиксировать *основные допущения* теории квантовых алгоритмов.

Алгоритм Дойча работает с квантовой системой $H_4 := H_2 \otimes H_2$ (напомним, что H_2 — это двумерное гильбертово пространство).

Вместо $|a\rangle \otimes |b\rangle$ пишем $|a\rangle|b\rangle$ и ссылаемся на $|a\rangle$ и соответственно $|b\rangle$ как на *содержимое* первого и соответственно второго регистра.

Дэвид Дойч в 1985 г. предложил квантовый алгоритм для решения описанной задачи, точнее, ее «квантовой» модификации.

Хотя алгоритм Дойча решает задачу, не представляющую никакого интереса, мы подробно рассмотрим его, чтобы на простом примере зафиксировать *основные допущения* теории квантовых алгоритмов.

Алгоритм Дойча работает с квантовой системой $H_4 := H_2 \otimes H_2$ (напомним, что H_2 — это двумерное гильбертово пространство).

Вместо $|a\rangle \otimes |b\rangle$ пишем $|a\rangle|b\rangle$ и ссылаемся на $|a\rangle$ и соответственно $|b\rangle$ как на *содержимое* первого и соответственно второго регистра.

Алгоритм Дойча

- 1 Инициализируем H_4 в состояние $|0\rangle|1\rangle$.
- 2 Применяем преобразование Адамара $R_4 := R_2 \otimes R_2$.
- 3 Применяем оператор $U_f: |a\rangle|b\rangle \mapsto |a\rangle|f(a) \oplus b\rangle$, где \oplus — сумма mod 2.
- 4 Применяем преобразование Адамара R_2 к первому регистру.
- 5 Измеряем первый регистр; если получится $|0\rangle$, то f — константа, если получится $|1\rangle$, то f — не константа.

Допущения теории квантовых алгоритмов

- 1 Инициализируем H_4 в состоянии $|0\rangle|1\rangle$.
- 2 Применяем преобразование Адамара R_4 .
- 3 Применяем оператор $U_f: |a\rangle|b\rangle \mapsto |a\rangle|f(a) \oplus b\rangle$.
- 4 Применяем преобразование Адамара R_2 к первому регистру.
- 5 Измеряем первый регистр; если получится $|0\rangle$, то f — константа, если получится $|1\rangle$, то f — не константа.

Допущения теории квантовых алгоритмов

- 1 Инициализируем H_4 в состоянии $|0\rangle|1\rangle$.
Можно «приготавливать» системы в некоторых заданных состояниях!
- 2 Применяем преобразование Адамара R_4 .
- 3 Применяем оператор $U_f: |a\rangle|b\rangle \mapsto |a\rangle|f(a) \oplus b\rangle$.
- 4 Применяем преобразование Адамара R_2 к первому регистру.
- 5 Измеряем первый регистр; если получится $|0\rangle$, то f — константа, если получится $|1\rangle$, то f — не константа.

Допущения теории квантовых алгоритмов

- 1 Инициализируем N_4 в состояние $|0\rangle|1\rangle$.
Можно «приготавливать» системы в некоторых заданных состояниях!
- 2 Применяем преобразование Адамара R_4 .
- 3 Применяем оператор $U_f: |a\rangle|b\rangle \mapsto |a\rangle|f(a) \oplus b\rangle$.
- 4 Применяем преобразование Адамара R_2 к первому регистру.
Можно применять к системам любые унитарные операторы!
- 5 Измеряем первый регистр; если получится $|0\rangle$, то f — константа, если получится $|1\rangle$, то f — не константа.

Допущения теории квантовых алгоритмов

- 1 Инициализируем H_4 в состояние $|0\rangle|1\rangle$.
Можно «приготавливать» системы в некоторых заданных состояниях!
- 2 Применяем преобразование Адамара R_4 .
- 3 Применяем оператор $U_f: |a\rangle|b\rangle \mapsto |a\rangle|f(a) \oplus b\rangle$.
- 4 Применяем преобразование Адамара R_2 к первому регистру.
Можно применять к системам любые унитарные операторы!
- 5 Измеряем первый регистр; если получится $|0\rangle$, то f — константа, если получится $|1\rangle$, то f — не константа.
Можно измерять системы, т.е. определять вероятность нахождения системы в некотором состоянии!

Допущения теории квантовых алгоритмов

- 1 Инициализируем N_4 в состояние $|0\rangle|1\rangle$.
Можно «приготавливать» системы в некоторых заданных состояниях!
- 2 Применяем преобразование Адамара R_4 .
- 3 Применяем оператор $U_f : |a\rangle|b\rangle \mapsto |a\rangle|f(a) \oplus b\rangle$.
- 4 Применяем преобразование Адамара R_2 к первому регистру.
Можно применять к системам любые унитарные операторы!
- 5 Измеряем первый регистр; если получится $|0\rangle$, то f — константа, если получится $|1\rangle$, то f — не константа.
Можно измерять системы, т.е. определять вероятность нахождения системы в некотором состоянии!

Мы подробно обсудим эти допущения немного позднее, а пока покажем, что алгоритм Дойча корректен.

Вычислим состояние системы на каждом шаге алгоритма Дойча.
После шага 1 состояние $|0\rangle|1\rangle$.

Вычислим состояние системы на каждом шаге алгоритма Дойча.

После шага 1 состояние $|0\rangle|1\rangle$.

На шаге 2 к каждому регистру применяется преобразование Адамара R_2 ,

т.е. оператор с матрицей $\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$.

Вычислим состояние системы на каждом шаге алгоритма Дойча.

После шага 1 состояние $|0\rangle|1\rangle$.

На шаге 2 к каждому регистру применяется преобразование Адамара R_2 ,

т.е. оператор с матрицей $\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$. Получится $\frac{1}{2}(|0\rangle + |1\rangle)(|0\rangle - |1\rangle)$.

Вычислим состояние системы на каждом шаге алгоритма Дойча.

После шага 1 состояние $|0\rangle|1\rangle$.

На шаге 2 к каждому регистру применяется преобразование Адамара R_2 ,

т.е. оператор с матрицей $\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$. Получится $\frac{1}{2}(|0\rangle + |1\rangle)(|0\rangle - |1\rangle)$.

Обозначив $|0\rangle - |1\rangle$ через $|\chi\rangle$, запишем результат как $\frac{1}{2}(|0\rangle + |1\rangle)|\chi\rangle$.

Вычислим состояние системы на каждом шаге алгоритма Дойча.

После шага 1 состояние $|0\rangle|1\rangle$.

На шаге 2 к каждому регистру применяется преобразование Адамара R_2 ,

т.е. оператор с матрицей $\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$. Получится $\frac{1}{2}(|0\rangle + |1\rangle)(|0\rangle - |1\rangle)$.

Обозначив $|0\rangle - |1\rangle$ через $|\chi\rangle$, запишем результат как $\frac{1}{2}(|0\rangle + |1\rangle)|\chi\rangle$.

На шаге 3 применяется оператор U_f . Это применение и играет роль единственного вызова функции f . Именно тут есть некоторый обман (в условии задачи говорилось, что можно обращаться к f , а не к U_f), так что молчаливо делается еще одно (очень сильное) допущение: **нужные нам функции продолжаются до соответствующих унитарных операторов.**

Вычислим состояние системы на каждом шаге алгоритма Дойча.

После шага 1 состояние $|0\rangle|1\rangle$.

На шаге 2 к каждому регистру применяется преобразование Адамара R_2 ,

т.е. оператор с матрицей $\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$. Получится $\frac{1}{2}(|0\rangle + |1\rangle)(|0\rangle - |1\rangle)$.

Обозначив $|0\rangle - |1\rangle$ через $|\chi\rangle$, запишем результат как $\frac{1}{2}(|0\rangle + |1\rangle)|\chi\rangle$.

На шаге 3 применяется оператор U_f . Это применение и играет роль единственного вызова функции f . Именно тут есть некоторый обман (в условии задачи говорилось, что можно обращаться к f , а не к U_f), так что молчаливо делается еще одно (очень сильное) допущение: **нужные нам функции продолжаются до соответствующих унитарных операторов.**

По определению $U_f|a\rangle|\chi\rangle = |a\rangle|f(a) \oplus \chi\rangle$.

Вычислим состояние системы на каждом шаге алгоритма Дойча.

После шага 1 состояние $|0\rangle|1\rangle$.

На шаге 2 к каждому регистру применяется преобразование Адамара R_2 ,

т.е. оператор с матрицей $\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$. Получится $\frac{1}{2}(|0\rangle + |1\rangle)(|0\rangle - |1\rangle)$.

Обозначив $|0\rangle - |1\rangle$ через $|\chi\rangle$, запишем результат как $\frac{1}{2}(|0\rangle + |1\rangle)|\chi\rangle$.

На шаге 3 применяется оператор U_f . Это применение и играет роль единственного вызова функции f . Именно тут есть некоторый обман (в условии задачи говорилось, что можно обращаться к f , а не к U_f), так что молчаливо делается еще одно (очень сильное) допущение: **нужные нам функции продолжаются до соответствующих унитарных операторов.**

По определению $U_f|a\rangle|\chi\rangle = |a\rangle|f(a) \oplus \chi\rangle$. Ясно, что $0 \oplus \chi = \chi$, а $1 \oplus \chi = |1\rangle - |0\rangle = -\chi$.

Вычислим состояние системы на каждом шаге алгоритма Дойча.

После шага 1 состояние $|0\rangle|1\rangle$.

На шаге 2 к каждому регистру применяется преобразование Адамара R_2 ,

т.е. оператор с матрицей $\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$. Получится $\frac{1}{2}(|0\rangle + |1\rangle)(|0\rangle - |1\rangle)$.

Обозначив $|0\rangle - |1\rangle$ через $|\chi\rangle$, запишем результат как $\frac{1}{2}(|0\rangle + |1\rangle)|\chi\rangle$.

На шаге 3 применяется оператор U_f . Это применение и играет роль единственного вызова функции f . Именно тут есть некоторый обман (в условии задачи говорилось, что можно обращаться к f , а не к U_f), так что молчаливо делается еще одно (очень сильное) допущение: **нужные нам функции продолжаются до соответствующих унитарных операторов.**

По определению $U_f|a\rangle|\chi\rangle = |a\rangle|f(a) \oplus \chi\rangle$. Ясно, что $0 \oplus \chi = \chi$, а $1 \oplus \chi = |1\rangle - |0\rangle = -\chi$. Поэтому $f(a) \oplus \chi = (-1)^{f(a)}\chi$.

Вычислим состояние системы на каждом шаге алгоритма Дойча.

После шага 1 состояние $|0\rangle|1\rangle$.

На шаге 2 к каждому регистру применяется преобразование Адамара R_2 ,

т.е. оператор с матрицей $\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$. Получится $\frac{1}{2}(|0\rangle + |1\rangle)(|0\rangle - |1\rangle)$.

Обозначив $|0\rangle - |1\rangle$ через $|\chi\rangle$, запишем результат как $\frac{1}{2}(|0\rangle + |1\rangle)|\chi\rangle$.

На шаге 3 применяется оператор U_f . Это применение и играет роль единственного вызова функции f . Именно тут есть некоторый обман (в условии задачи говорилось, что можно обращаться к f , а не к U_f), так что молчаливо делается еще одно (очень сильное) допущение: **нужные нам функции продолжаются до соответствующих унитарных операторов.**

По определению $U_f|a\rangle|\chi\rangle = |a\rangle|f(a) \oplus \chi\rangle$. Ясно, что $0 \oplus \chi = \chi$, а $1 \oplus \chi = |1\rangle - |0\rangle = -\chi$. Поэтому $f(a) \oplus \chi = (-1)^{f(a)}\chi$. Получаем

$$U_f \frac{1}{2}(|0\rangle + |1\rangle)|\chi\rangle = \frac{1}{2} \left((-1)^{f(0)}|0\rangle|\chi\rangle + (-1)^{f(1)}|1\rangle|\chi\rangle \right).$$

На шаге 4 к первому регистру применяется преобразование Адамара.

На шаге 4 к первому регистру применяется преобразование Адамара.

Учитывая, что $\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle + |1\rangle$, а $\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |0\rangle - |1\rangle$, получаем

$$\begin{aligned} & \frac{1}{2\sqrt{2}} \left((-1)^{f(0)}(|0\rangle + |1\rangle)|\chi\rangle + (-1)^{f(1)}(|0\rangle - |1\rangle)|\chi\rangle \right) = \\ & = \frac{1}{2\sqrt{2}} \left(\left((-1)^{f(0)} + (-1)^{f(1)} \right) |0\rangle|\chi\rangle + \left((-1)^{f(0)} - (-1)^{f(1)} \right) |1\rangle|\chi\rangle \right) = \\ & = \begin{cases} \pm |0\rangle \frac{1}{\sqrt{2}} |\chi\rangle, & \text{если } f \text{ константа,} \\ \pm |1\rangle \frac{1}{\sqrt{2}} |\chi\rangle, & \text{если } f \text{ не константа.} \end{cases} \end{aligned}$$

На шаге 4 к первому регистру применяется преобразование Адамара.

Учитывая, что $\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle + |1\rangle$, а $\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |0\rangle - |1\rangle$, получаем

$$\begin{aligned} & \frac{1}{2\sqrt{2}} \left((-1)^{f(0)}(|0\rangle + |1\rangle)|\chi\rangle + (-1)^{f(1)}(|0\rangle - |1\rangle)|\chi\rangle \right) = \\ & = \frac{1}{2\sqrt{2}} \left(\left((-1)^{f(0)} + (-1)^{f(1)} \right) |0\rangle|\chi\rangle + \left((-1)^{f(0)} - (-1)^{f(1)} \right) |1\rangle|\chi\rangle \right) = \\ & = \begin{cases} \pm |0\rangle \frac{1}{\sqrt{2}} |\chi\rangle, & \text{если } f \text{ константа,} \\ \pm |1\rangle \frac{1}{\sqrt{2}} |\chi\rangle, & \text{если } f \text{ не константа.} \end{cases} \end{aligned}$$

Теперь понятно, что, замерив на шаге 5 значение первого регистра, получим $|0\rangle$, если f — константа, и $|1\rangle$, если f — не константа.

Вернемся к сделанным допущениям.

Допущения теории квантовых алгоритмов

- 1 Можно «приготавливать» системы в некоторых заданных состояниях.
- 2 Можно применять к системам любые унитарные операторы.
- 3 Можно определять вероятность нахождения системы в некотором состоянии.

Вернемся к сделанным допущениям.

Допущения теории квантовых алгоритмов

- 1 Можно «приготавливать» системы в некоторых заданных состояниях.
- 2 Можно применять к системам любые унитарные операторы.
- 3 Можно определять вероятность нахождения системы в некотором состоянии.

С точки зрения квантовой механики эти допущения выглядят разумно в том смысле, что они не противоречат никаким ее выводам.

Вернемся к сделанным допущениям.

Допущения теории квантовых алгоритмов

- 1 Можно «приготавливать» системы в некоторых заданных состояниях.
- 2 Можно применять к системам любые унитарные операторы.
- 3 Можно определять вероятность нахождения системы в некотором состоянии.

С точки зрения квантовой механики эти допущения выглядят разумно в том смысле, что они не противоречат никаким ее выводам.

Вопрос о том, насколько эти допущения *реализуемы*, выходит за рамки нашего курса — это вопрос к инженерам-физикам.

Вернемся к сделанным допущениям.

Допущения теории квантовых алгоритмов

- 1 Можно «приготавливать» системы в некоторых заданных состояниях.
- 2 Можно применять к системам любые унитарные операторы.
- 3 Можно определять вероятность нахождения системы в некотором состоянии.

С точки зрения квантовой механики эти допущения выглядят разумно в том смысле, что они не противоречат никаким ее выводам.

Вопрос о том, насколько эти допущения *реализуемы*, выходит за рамки нашего курса — это вопрос к инженерам-физикам.

Важно понимать, что «квантовое превосходство», т.е. способность квантовых алгоритмов решать какие-то задачи «намного быстрее», чем это делают обычные алгоритмы, основано на двух подменах понятий:

Вернемся к сделанным допущениям.

Допущения теории квантовых алгоритмов

- 1 Можно «приготавливать» системы в некоторых заданных состояниях.
- 2 Можно применять к системам любые унитарные операторы.
- 3 Можно определять вероятность нахождения системы в некотором состоянии.

С точки зрения квантовой механики эти допущения выглядят разумно в том смысле, что они не противоречат никаким ее выводам.

Вопрос о том, насколько эти допущения *реализуемы*, выходит за рамки нашего курса — это вопрос к инженерам-физикам.

Важно понимать, что «квантовое превосходство», т.е. способность квантовых алгоритмов решать какие-то задачи «намного быстрее», чем это делают обычные алгоритмы, основано на двух подменах понятий: во-первых, квантовые алгоритмы решают **не те же самые** задачи (замена функции f на оператор U_f в алгоритме Дойча)

Вернемся к сделанным допущениям.

Допущения теории квантовых алгоритмов

- 1 Можно «приготавливать» системы в некоторых заданных состояниях.
- 2 Можно применять к системам любые унитарные операторы.
- 3 Можно определять вероятность нахождения системы в некотором состоянии.

С точки зрения квантовой механики эти допущения выглядят разумно в том смысле, что они не противоречат никаким ее выводам.

Вопрос о том, насколько эти допущения *реализуемы*, выходит за рамки нашего курса — это вопрос к инженерам-физикам.

Важно понимать, что «квантовое превосходство», т.е. способность квантовых алгоритмов решать какие-то задачи «намного быстрее», чем это делают обычные алгоритмы, основано на двух подменах понятий: во-первых, квантовые алгоритмы решают **не те же самые** задачи (замена функции f на оператор U_f в алгоритме Дойча), во-вторых, число шагов считается **иначе** (каждое применение унитарного оператора рассматривается как один шаг вне зависимости от размерности пространства H).

В 1992 г. Дойч и Ричард Йожа обобщили алгоритм Дойча.

В 1992 г. Дойч и Ричард Йожа обобщили алгоритм Дойча.

Рассматриваются функции $f: \mathbb{F}^n \rightarrow \mathbb{F}$, где n произвольно.

В 1992 г. Дойч и Ричард Йожа обобщили алгоритм Дойча.

Рассматриваются функции $f: \mathbb{F}^n \rightarrow \mathbb{F}$, где n произвольно.

В задаче заранее известно, что f — либо константа, либо *сбалансированная* функция, т.е. принимает значения 0 и 1 одинаковое число раз (по 2^{n-1} раз).

В 1992 г. Дойч и Ричард Йожа обобщили алгоритм Дойча.

Рассматриваются функции $f: \mathbb{F}^n \rightarrow \mathbb{F}$, где n произвольно.

В задаче заранее известно, что f — либо константа, либо *сбалансированная* функция, т.е. принимает значения 0 и 1 одинаковое число раз (по 2^{n-1} раз).

Сама функция f неизвестна, но к ней можно обращаться как к черному ящику: задать аргумент $x \in \{0, 1\}^n$ и получить значение $f(x)$.

В 1992 г. Дойч и Ричард Йожа обобщили алгоритм Дойча.

Рассматриваются функции $f: \mathbb{F}^n \rightarrow \mathbb{F}$, где n произвольно.

В задаче заранее известно, что f — либо константа, либо *сбалансированная* функция, т.е. принимает значения 0 и 1 одинаковое число раз (по 2^{n-1} раз).

Сама функция f неизвестна, но к ней можно обращаться как к черному ящику: задать аргумент $x \in \{0, 1\}^n$ и получить значение $f(x)$.

Нужно узнать, является функция f константой.

В 1992 г. Дойч и Ричард Йожа обобщили алгоритм Дойча.

Рассматриваются функции $f: \mathbb{F}^n \rightarrow \mathbb{F}$, где n произвольно.

В задаче заранее известно, что f — либо константа, либо *сбалансированная* функция, т.е. принимает значения 0 и 1 одинаковое число раз (по 2^{n-1} раз).

Сама функция f неизвестна, но к ней можно обращаться как к черному ящику: задать аргумент $x \in \{0, 1\}^n$ и получить значение $f(x)$.

Нужно узнать, является функция f константой.

Ясно, что любой детерминированный алгоритм для этой задачи требует $2^{n-1} + 1$ вызовов функции f в наихудшем случае.

Алгоритм Дойча–Йожи работает с квантовой системой $H := H_2^n \otimes H_2$ (напомним, что $H_2^n = \underbrace{H_2 \otimes \cdots \otimes H_2 \otimes H_2}_{n \text{ раз}}$ — это n -я тензорная степень двумерного гильбертова пространства H_2).

Алгоритм Дойча–Йожи работает с квантовой системой $H := H_{2^n} \otimes H_2$ (напомним, что $H_{2^n} = \underbrace{H_2 \otimes \cdots \otimes H_2 \otimes H_2}_{n \text{ раз}}$ — это n -я тензорная степень

двумерного гильбертова пространства H_2).

Вместо $|a\rangle \otimes |b\rangle$ пишем $|a\rangle|b\rangle$ и ссылаемся на $|a\rangle$ и соответственно $|b\rangle$ как на *содержимое* первого и соответственно второго регистра.

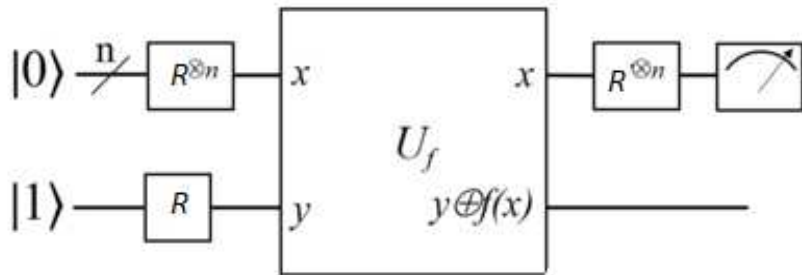
Алгоритм Дойча–Йожи работает с квантовой системой $H := H_{2^n} \otimes H_2$ (напомним, что $H_{2^n} = \underbrace{H_2 \otimes \dots \otimes H_2}_{n \text{ раз}} \otimes H_2$ — это n -я тензорная степень

двумерного гильбертова пространства H_2).

Вместо $|a\rangle \otimes |b\rangle$ пишем $|a\rangle|b\rangle$ и ссылаемся на $|a\rangle$ и соответственно $|b\rangle$ как на *содержимое* первого и соответственно второго регистра.

Алгоритм Дойча–Йожи

- 1 Инициализируем H в состояние $|0\rangle|1\rangle$.
- 2 Применяем преобразование Адамара $R_{2^{n+1}} := R_{2^n} \otimes R_2$.
- 3 Применяем оператор $U_f: |a\rangle|b\rangle \mapsto |a\rangle|f(a) \oplus b\rangle$, где \oplus — сумма mod 2.
- 4 Применяем преобразование Адамара R_{2^n} к первому регистру.
- 5 Измеряем первый регистр; если получится $|0\rangle$, то f — константа, если получится какой-нибудь другой базисный вектор $|u\rangle$, $u \neq 0$, то f — сбалансированная функция.



Здесь (и во всех других алгоритмах) нужен явный вид оператора Адамара

$R_{2^n} := \underbrace{R_2 \otimes \cdots \otimes R_2 \otimes R_2}_{n \text{ раз}}$. Напомним, что $R_2 := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$, откуда

$R_2|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, а $R_2|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$.

Здесь (и во всех других алгоритмах) нужен явный вид оператора Адамара

$R_{2^n} := \underbrace{R_2 \otimes \cdots \otimes R_2 \otimes R_2}_{n \text{ раз}}$. Напомним, что $R_2 := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$, откуда

$$R_2|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \text{ а } R_2|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle).$$

Лемма

$$R_{2^n}|k\rangle = \frac{1}{\sqrt{2^n}} \sum_{u=0}^{2^n-1} (-1)^{k \cdot u} |u\rangle, \text{ где } k \cdot u := \sum_{i=0}^{n-1} k_i u_i \pmod{2}.$$

Здесь (и во всех других алгоритмах) нужен явный вид оператора Адамара

$R_{2^n} := \underbrace{R_2 \otimes \cdots \otimes R_2 \otimes R_2}_{n \text{ раз}}$. Напомним, что $R_2 := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$, откуда

$$R_2|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \text{ а } R_2|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle).$$

Лемма

$$R_{2^n}|k\rangle = \frac{1}{\sqrt{2^n}} \sum_{u=0}^{2^n-1} (-1)^{k \cdot u} |u\rangle, \text{ где } k \cdot u := \sum_{i=0}^{n-1} k_i u_i \pmod{2}.$$

Доказательство. Имеем $|k\rangle = |k_{n-1}\rangle \otimes \cdots \otimes |k_0\rangle$, откуда

$$\begin{aligned} R_{2^n}|k\rangle &= R_2|k_{n-1}\rangle \otimes \cdots \otimes R_2|k_0\rangle = \\ &= \frac{1}{\sqrt{2^n}} \left(|0\rangle + (-1)^{k_{n-1}} |1\rangle \right) \otimes \cdots \otimes \left(|0\rangle + (-1)^{k_0} |1\rangle \right). \end{aligned}$$

Здесь (и во всех других алгоритмах) нужен явный вид оператора Адамара

$R_{2^n} := \underbrace{R_2 \otimes \cdots \otimes R_2 \otimes R_2}_{n \text{ раз}}$. Напомним, что $R_2 := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$, откуда

$$R_2|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \text{ а } R_2|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle).$$

Лемма

$$R_{2^n}|k\rangle = \frac{1}{\sqrt{2^n}} \sum_{u=0}^{2^n-1} (-1)^{k \cdot u} |u\rangle, \text{ где } k \cdot u := \sum_{i=0}^{n-1} k_i u_i \pmod{2}.$$

Доказательство. Имеем $|k\rangle = |k_{n-1}\rangle \otimes \cdots \otimes |k_0\rangle$, откуда

$$\begin{aligned} R_{2^n}|k\rangle &= R_2|k_{n-1}\rangle \otimes \cdots \otimes R_2|k_0\rangle = \\ &= \frac{1}{\sqrt{2^n}} \left(|0\rangle + (-1)^{k_{n-1}} |1\rangle \right) \otimes \cdots \otimes \left(|0\rangle + (-1)^{k_0} |1\rangle \right). \end{aligned}$$

Раскрывая скобки, получим сумму 2^n слагаемых, каждое из которых определяется выбором $|0\rangle$ или $|1\rangle$ из каждого множителя, т.е. равно $(-1)^{k_{n-1}u_{n-1} \oplus \cdots \oplus k_0 u_0} |u_{n-1}\rangle \otimes \cdots \otimes |u_0\rangle$, где $u_i = 0$, если из i -го множителя выбран $|0\rangle$, и $u_i = 1$, если из i -го множителя выбран $|1\rangle$.

Здесь (и во всех других алгоритмах) нужен явный вид оператора Адамара

$R_{2^n} := \underbrace{R_2 \otimes \cdots \otimes R_2}_{n \text{ раз}} \otimes R_2$. Напомним, что $R_2 := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$, откуда

$$R_2|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \text{ а } R_2|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle).$$

Лемма

$$R_{2^n}|k\rangle = \frac{1}{\sqrt{2^n}} \sum_{u=0}^{2^n-1} (-1)^{k \cdot u} |u\rangle, \text{ где } k \cdot u := \sum_{i=0}^{n-1} k_i u_i \pmod{2}.$$

Доказательство. Имеем $|k\rangle = |k_{n-1}\rangle \otimes \cdots \otimes |k_0\rangle$, откуда

$$\begin{aligned} R_{2^n}|k\rangle &= R_2|k_{n-1}\rangle \otimes \cdots \otimes R_2|k_0\rangle = \\ &= \frac{1}{\sqrt{2^n}} \left(|0\rangle + (-1)^{k_{n-1}} |1\rangle \right) \otimes \cdots \otimes \left(|0\rangle + (-1)^{k_0} |1\rangle \right). \end{aligned}$$

Раскрывая скобки, получим сумму 2^n слагаемых, каждое из которых определяется выбором $|0\rangle$ или $|1\rangle$ из каждого множителя, т.е. равно $(-1)^{k_{n-1}u_{n-1} \oplus \cdots \oplus k_0 u_0} |u_{n-1}\rangle \otimes \cdots \otimes |u_0\rangle$, где $u_i = 0$, если из i -го множителя выбран $|0\rangle$, и $u_i = 1$, если из i -го множителя выбран $|1\rangle$. Если положить $|u\rangle := |u_{n-1}\rangle \otimes \cdots \otimes |u_0\rangle$, то это слагаемое можно записать как $(-1)^{k \cdot u} |u\rangle$.

Если в формулу

$$R_{2^n}|k\rangle = \frac{1}{\sqrt{2^n}} \sum_{u=0}^{2^n-1} (-1)^{k \cdot u} |u\rangle, \quad \text{где } k \cdot u := \sum_{i=0}^{n-1} k_i u_i \pmod{2}.$$

подставить $k = 0$, получим

$$R_{2^n}|0\rangle = \frac{1}{\sqrt{2^n}} \sum_{u=0}^{2^n-1} |u\rangle.$$

Если в формулу

$$R_{2^n}|k\rangle = \frac{1}{\sqrt{2^n}} \sum_{u=0}^{2^n-1} (-1)^{k \cdot u} |u\rangle, \quad \text{где } k \cdot u := \sum_{i=0}^{n-1} k_i u_i \pmod{2}.$$

подставить $k = 0$, получим

$$R_{2^n}|0\rangle = \frac{1}{\sqrt{2^n}} \sum_{u=0}^{2^n-1} |u\rangle.$$

Это сумма всех базисных векторов пространства H_{2^n} с одинаковыми коэффициентами, иначе говоря, вектор, в котором равномерно представлены все базисные вектора.

Если в формулу

$$R_{2^n}|k\rangle = \frac{1}{\sqrt{2^n}} \sum_{u=0}^{2^n-1} (-1)^{k \cdot u} |u\rangle, \quad \text{где } k \cdot u := \sum_{i=0}^{n-1} k_i u_i \pmod{2}.$$

подставить $k = 0$, получим

$$R_{2^n}|0\rangle = \frac{1}{\sqrt{2^n}} \sum_{u=0}^{2^n-1} |u\rangle.$$

Это сумма всех базисных векторов пространства H_{2^n} с одинаковыми коэффициентами, иначе говоря, вектор, в котором равномерно представлены все базисные вектора.

Применяя какой-то оператор к этому вектору, мы неявно применяем этот оператор *ко всем базисным векторам одновременно!*

Вычислим состояние системы на каждом шаге алгоритма Дойча–Йожи.
После шага 1 состояние $|0\rangle|1\rangle$.

Вычислим состояние системы на каждом шаге алгоритма Дойча–Йожи.
После шага 1 состояние $|0\rangle|1\rangle$.
На шаге 2 к каждому регистру применяется преобразование Адамара.

Вычислим состояние системы на каждом шаге алгоритма Дойча–Йожи.

После шага 1 состояние $|0\rangle|1\rangle$.

На шаге 2 к каждому регистру применяется преобразование Адамара.

Получится $\frac{1}{\sqrt{2^{n+1}}} \left(\sum_{u=0}^{2^n-1} |u\rangle \right) |\chi\rangle$, где $|\chi\rangle := |0\rangle - |1\rangle$.

Вычислим состояние системы на каждом шаге алгоритма Дойча–Йожи.

После шага 1 состояние $|0\rangle|1\rangle$.

На шаге 2 к каждому регистру применяется преобразование Адамара.

Получится $\frac{1}{\sqrt{2^{n+1}}} \left(\sum_{u=0}^{2^n-1} |u\rangle \right) |\chi\rangle$, где $|\chi\rangle := |0\rangle - |1\rangle$.

На шаге 3 применяется оператор U_f . Получим

$\frac{1}{\sqrt{2^{n+1}}} \left(\sum_{u=0}^{2^n-1} (-1)^{f(u)} |u\rangle \right) |\chi\rangle$.

Вычислим состояние системы на каждом шаге алгоритма Дойча–Йожи.

После шага 1 состояние $|0\rangle|1\rangle$.

На шаге 2 к каждому регистру применяется преобразование Адамара.

Получится $\frac{1}{\sqrt{2^{n+1}}} \left(\sum_{u=0}^{2^n-1} |u\rangle \right) |\chi\rangle$, где $|\chi\rangle := |0\rangle - |1\rangle$.

На шаге 3 применяется оператор U_f . Получим

$\frac{1}{\sqrt{2^{n+1}}} \left(\sum_{u=0}^{2^n-1} (-1)^{f(u)} |u\rangle \right) |\chi\rangle$.

На шаге 4 к первому регистру применяется преобразование Адамара.

Получим

$$\frac{1}{2^n \sqrt{2}} \left(\sum_{u=0}^{2^n-1} \sum_{j=0}^{2^n-1} (-1)^{u \cdot j} |j\rangle (-1)^{f(u)} |\chi\rangle \right) =$$

$$\frac{1}{2^n \sqrt{2}} \sum_{j=0}^{2^n-1} |j\rangle \sum_{u=0}^{2^n-1} (-1)^{u \cdot j} (-1)^{f(u)} |\chi\rangle.$$

Разберем, чему равно выражение

$$\frac{1}{2^n \sqrt{2}} \sum_{j=0}^{2^n-1} |j\rangle \sum_{u=0}^{2^n-1} (-1)^{u \cdot j} (-1)^{f(u)} |\chi\rangle \quad (*)$$

в зависимости от того, какова функция f . Напомним, что по условию задачи f — либо константа, либо сбалансированная функция.

Разберем, чему равно выражение

$$\frac{1}{2^n \sqrt{2}} \sum_{j=0}^{2^n-1} |j\rangle \sum_{u=0}^{2^n-1} (-1)^{u \cdot j} (-1)^{f(u)} |\chi\rangle \quad (*)$$

в зависимости от того, какова функция f . Напомним, что по условию задачи f — либо константа, либо сбалансированная функция.

Если f — константа, то $(-1)^{f(u)}$ от u не зависит и

$$\sum_{u=0}^{2^n-1} (-1)^{u \cdot j} (-1)^{f(u)} = \begin{cases} \pm 2^n, & \text{если } j = 0, \\ 0, & \text{если } j \neq 0. \end{cases}$$

Разберем, чему равно выражение

$$\frac{1}{2^n \sqrt{2}} \sum_{j=0}^{2^n-1} |j\rangle \sum_{u=0}^{2^n-1} (-1)^{u \cdot j} (-1)^{f(u)} |\chi\rangle \quad (*)$$

в зависимости от того, какова функция f . Напомним, что по условию задачи f — либо константа, либо сбалансированная функция.

Если f — константа, то $(-1)^{f(u)}$ от u не зависит и

$$\sum_{u=0}^{2^n-1} (-1)^{u \cdot j} (-1)^{f(u)} = \begin{cases} \pm 2^n, & \text{если } j = 0, \\ 0, & \text{если } j \neq 0. \end{cases}$$

Почему $\sum_{u=0}^{2^n-1} (-1)^{u \cdot j} = 0$ при $j \neq 0$? Напомним, $u \cdot j = \sum_{i=0}^{n-1} u_i j_i \pmod{2}$.

Разберем, чему равно выражение

$$\frac{1}{2^n \sqrt{2}} \sum_{j=0}^{2^n-1} |j\rangle \sum_{u=0}^{2^n-1} (-1)^{u \cdot j} (-1)^{f(u)} |\chi\rangle \quad (*)$$

в зависимости от того, какова функция f . Напомним, что по условию задачи f — либо константа, либо сбалансированная функция.

Если f — константа, то $(-1)^{f(u)}$ от u не зависит и

$$\sum_{u=0}^{2^n-1} (-1)^{u \cdot j} (-1)^{f(u)} = \begin{cases} \pm 2^n, & \text{если } j = 0, \\ 0, & \text{если } j \neq 0. \end{cases}$$

Почему $\sum_{u=0}^{2^n-1} (-1)^{u \cdot j} = 0$ при $j \neq 0$? Напомним, $u \cdot j = \sum_{i=0}^{n-1} u_i j_i \pmod{2}$. Рассмотрим линейное уравнение $j \cdot x = \sum_{i=0}^{n-1} j_i x_i = 0$ в пространстве \mathbb{F}^n . При ненулевом j пространство решений этого уравнения имеет размерность $n - 1$, откуда в \mathbb{F}^n есть 2^{n-1} таких векторов u , что $u \cdot j = 0$, и 2^{n-1} таких, что $u \cdot j = 1$.

Разберем, чему равно выражение

$$\frac{1}{2^n \sqrt{2}} \sum_{j=0}^{2^n-1} |j\rangle \sum_{u=0}^{2^n-1} (-1)^{u \cdot j} (-1)^{f(u)} |\chi\rangle \quad (*)$$

в зависимости от того, какова функция f . Напомним, что по условию задачи f — либо константа, либо сбалансированная функция.

Если f — константа, то $(-1)^{f(u)}$ от u не зависит и

$$\sum_{u=0}^{2^n-1} (-1)^{u \cdot j} (-1)^{f(u)} = \begin{cases} \pm 2^n, & \text{если } j = 0, \\ 0, & \text{если } j \neq 0. \end{cases}$$

Почему $\sum_{u=0}^{2^n-1} (-1)^{u \cdot j} = 0$ при $j \neq 0$? Напомним, $u \cdot j = \sum_{i=0}^{n-1} u_i j_i \pmod{2}$. Рассмотрим линейное уравнение $j \cdot x = \sum_{i=0}^{n-1} j_i x_i = 0$ в пространстве \mathbb{F}^n . При ненулевом j пространство решений этого уравнения имеет размерность $n - 1$, откуда в \mathbb{F}^n есть 2^{n-1} таких векторов u , что $u \cdot j = 0$, и 2^{n-1} таких, что $u \cdot j = 1$. Итак, при $j \neq 0$ в сумме $\sum_{u=0}^{2^n-1} (-1)^{u \cdot j}$ равное число $+1$ и -1 .

Разберем, чему равно выражение

$$\frac{1}{2^n \sqrt{2}} \sum_{j=0}^{2^n-1} |j\rangle \sum_{u=0}^{2^n-1} (-1)^{u \cdot j} (-1)^{f(u)} |\chi\rangle \quad (*)$$

в зависимости от того, какова функция f . Напомним, что по условию задачи f — либо константа, либо сбалансированная функция.

Если f — константа, то $(-1)^{f(u)}$ от u не зависит и

$$\sum_{u=0}^{2^n-1} (-1)^{u \cdot j} (-1)^{f(u)} = \begin{cases} \pm 2^n, & \text{если } j = 0, \\ 0, & \text{если } j \neq 0. \end{cases}$$

Почему $\sum_{u=0}^{2^n-1} (-1)^{u \cdot j} = 0$ при $j \neq 0$? Напомним, $u \cdot j = \sum_{i=0}^{n-1} u_i j_i \pmod{2}$. Рассмотрим линейное уравнение $j \cdot x = \sum_{i=0}^{n-1} j_i x_i = 0$ в пространстве \mathbb{F}^n . При ненулевом j пространство решений этого уравнения имеет размерность $n - 1$, откуда в \mathbb{F}^n есть 2^{n-1} таких векторов u , что $u \cdot j = 0$, и 2^{n-1} таких, что $u \cdot j = 1$. Итак, при $j \neq 0$ в сумме $\sum_{u=0}^{2^n-1} (-1)^{u \cdot j}$ равное число $+1$ и -1 .

Следовательно, если f — константа, то выражение (*) равно $\pm |0\rangle \frac{1}{\sqrt{2}} |\chi\rangle$, и потому *измерение первого регистра на шаге 5 даст $|0\rangle$ с вероятностью 1.*

Если f — сбалансированная функция, то среди чисел $(-1)^{f(u)}$ равное количество $+1$ и -1 .

Если f — сбалансированная функция, то среди чисел $(-1)^{f(u)}$ равное количество $+1$ и -1 . Поэтому $\sum_{u=0}^{2^n-1} (-1)^{u \cdot j} (-1)^{f(u)} = 0$, если $j \neq 0$.

Если f — сбалансированная функция, то среди чисел $(-1)^{f(u)}$ равное количество $+1$ и -1 . Поэтому $\sum_{u=0}^{2^n-1} (-1)^{u \cdot j} (-1)^{f(u)} = 0$, если $j \neq 0$. Следовательно, если f — сбалансированная функция, то в выражении (*) пропадёт слагаемое с $j = 0$ и останется

$$\frac{1}{2^n \sqrt{2}} \sum_{j=1}^{2^n-1} |j\rangle \sum_{u=0}^{2^n-1} (-1)^{u \cdot j} (-1)^{f(u)} |\chi\rangle.$$

Если f — сбалансированная функция, то среди чисел $(-1)^{f(u)}$ равное количество $+1$ и -1 . Поэтому $\sum_{u=0}^{2^n-1} (-1)^{u \cdot j} (-1)^{f(u)} = 0$, если $j \neq 0$. Следовательно, если f — сбалансированная функция, то в выражении (*) пропадёт слагаемое с $j = 0$ и останется

$$\frac{1}{2^n \sqrt{2}} \sum_{j=1}^{2^n-1} |j\rangle \sum_{u=0}^{2^n-1} (-1)^{u \cdot j} (-1)^{f(u)} |\chi\rangle.$$

Измерение первого регистра на шаге 5 даст какой-то вектор $|j\rangle$ с $j \neq 0$.