§ 2.7. **Сложность алгоритмов.**

Определение: Однородный класс вычислительных задач будем называть «проблемой (массовой задачей)» и обозначать через **T,** а алгоритм, ее решающий, через **A**. Частный случай **T** обозначим через $I \in T$ **. A** выполняет последовательность вычислений S_I. Длина S_I характеризует время вычислений. Глубина S_I - число уровней параллельных шагов – время параллельных вычислений.

1) $\mathrm{F_A}(n) = \max\{\mu(S_I) / I \in \mathrm{T}, |I| = n\}$ сложность вычислений «в наихудшем случае», где μ - мера вычисления может быть выбрана как: а) число элементов или б) глубина или в) число модулей (уровень технологии); $|I|$ – размер.

2) $\mathrm{M_A}(n) = \sum_{I \in T_n} P(I) \cdot \mu(S_I)$ сложность «поведения в среднем», где $P(I)$ - вероятность появления I среди возможных частных случаев T_n.

3) Анализ алгоритмов связан с вопросом «для заданной функции размера $|I|$ и меры вычисления $\mu(S_I)$ точно определить для данного алгоритма **A,** решающего проблему **T,** либо сложность $\mathrm{F_A}(n)$ «для наихудшего случая», либо, при подходящих предположениях, «поведение в среднем» $\mathrm{M_A}(n)$. (Кнут «Искусство программирования»).

4) **Сложность задачи** – сложность наилучшего алгоритма, известного для ее решения (нижняя оценка сложности алгоритма).

Определение: Будем говорить, что функция $f(n)$ есть $O(g(n))$, если существует константа c такая, что $|f(n)| \le c \cdot g(n)$ для всех натуральных n.

Основной вопрос теории сложности: с какой стоимостью может быть решена заданная проблема?

2.7.1. **Классификация задач по степени сложности**

а) линейные $O(n)$; б) быстрее их - $O(\log_2 n)$; в) полиномиальные алгоритмы принадлежат к классу **P**, для которых временная сложность порядка $O(n^k)$, где $k > 0$ – целое;

г) экспоненциальные – для них не существует оценки $O(n^k)$.

Определение: Задача «труднорешаема», если для нее не существует «полиномиального» алгоритма.

Замечание: для малых n «экспоненциальный» алгоритм может быть быстрее полиномиального.

Класс P:

1) найти эйлеров цикл на графе из m ребер: алгоритм проверки циклов - $O(m)$; 2) задача Прима-Краскала: n городов в плоской стране. Общая длина телефонных линий соединения городов минимальна. Жадный алгоритм находит «остовное» дерево за $O(\log_2 n)$.

3) Быстрое преобразование Фурье (БПФ) за $O(\log_2 n)$ шагов; 4) Умножение целых чисел по алгоритму Шенхаге-Штрассена (Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов.-М.: Мир, 1979.-536 с.) за $O(n \log n \cdot \log\log n)$ шагов по сравнению с умножением двух n – разрядных чисел по традиционному алгоритму $O(n^2)$.

5) Умножение матриц - $O(n^3)$.

КЛАСС E:

1) построить множество всех подмножеств; 2) все полные подграфы графа или все поддеревья графа.

ЗАДАЧИ, не принадлежащие к классу **P** и не принадлежащие к классу **E :**

1) задача **«коммивояжера»;** 2) задача **«составления расписаний при определенных условиях»;** 3) задача **«оптимальной загрузки емкости (рюкзака, вагонов поезда, самолета и т.д.);** 4) задача **«распознавания простого числа» и другие.**

Определение: Детерминированный алгоритм – алгоритм, в котором переход из одного состояния в другое однозначный. Недетерминированный имеет несколько вариантов перехода (вероятностный переход).

Определение: **Класс NP** задач, которые решаемы недетерминированными алгоритмами за время $O(n^k)$, то есть $P \subseteq NP$.

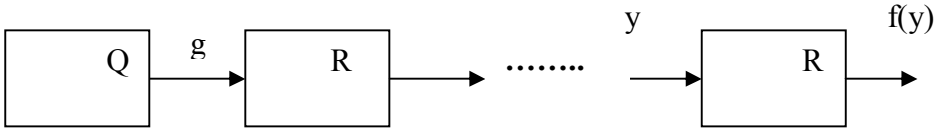
Замечание: Так как число путей вычисления может быть экспоненциально, то алгоритмы **NP** сильнее алгоритмов класса **P.**

Замечание: Оптимизация задачи «коммивояжера» - задача класса **NP,** так как полиномиального алгоритма пока нет. Алгоритм **NP** состоит из 2-х стадий: а) стадия угадывания последовательности городов; б) стадия проверки маршрута длины – полиномиальная процедура.

NP-трудные и NP-полные задачи

Определение: Задача **Q** ***полиномиально*** сводится к задаче **R** тогда и только тогда, когда выполнены условия:

1) Существуют функции $g(x)$ и $f(x)$, вычисляемые за ***полиномиальное*** время $O(n^k)$;
2) Для любого входа x и для любого частного случая задачи **Q** значение $g(x)$ - вход частного случая задачи **R;**
3) Для любого решения (выхода) y задачи **R** значение $f(y)$ - решение задачи **Q**:



Определение: Если одновременно задача **Q** ***полиномиально*** сводится к задаче **R** и задача **R** ***полиномиально*** сводится к задаче **Q**, то задачи **Q** и **R** ***полиномиально эквивалентны***.

Определение: Задача является **NP-трудной (или NP-сложной),** если каждая задача из класса **NP** ***полиномиально*** сводится к ней. Задача является **NP-полной,** если она входит в класс **NP** и является **NP-трудной.** Другими словами, задача **T** является **NP-трудной,** если она по крайней мере так сложна, как любая задача в **NP.**

NP-полные задачи – это самые трудные из **NP.**

Любая **NP**-полная задача **T** принадлежит **NP\P.** Точнее, задача **T** принадлежит к классу **P** тогда и только тогда, когда **P=NP.**

Теорема Кука (задача о выполнимости является NP-полной): **F** – формула из теории **L** (**ИВ** – исчисление высказываний) представлена в КНФ. Существует ли такое распределение истинностных значений высказывательных переменных, при которых формула **F выполнима?**

Доказательство: Обозначим задачу распределения истинностных значений высказывательных переменных, при которых формула **F выполнима**, через задачу **T**. Задача о выполнимости **T** полиномиально сводится к любой **NP-трудной** задаче**,** принадлежащей к классу **NP,** то есть она является **NP-полной.**

К настоящему времени установлена **NP**-полнота большого числа задач. Выше были перечислены некоторые задачи, которые не попадают ни в класс **P**, ни в класс **E**. Все они являются **NP**-полными.

Проблема состоит в следующем: можем ли мы надеяться, что какая-либо из этих задач имеет полиномиальную сложность?

По-видимому, ответ будет неудовлетворительным. Очень важным аргументом для такого вывода служит тот факт, что все задачи эквивалентны по сложности – стоит нам найти какой-то полиномиальный алгоритм для одной из этих задач, то все эти задачи становятся полиномиально сложны.