Іспит з теорії алгоритмів

ПМі-23

- Черненюк Катерина

Білет № 1

1) Базові поняття алгоритмів та їх складності. Оцінювання алгоритмів

Для оцінки алгоритму використовують такі критерії:

- розмір (число, що виражає обсяг вхідних даних, які потрібні для опису задачі)
- час (це часова складність алгоритму, а саме час, затрачений на виконання цього алгоритму)
- асимптотична складність (це відношення часової складності до збільшення розмірів задачі)
- ємнісна складність (затрачений обсяг пам'яті, необхідний для реалізації алгоритму)
- асимптотична ємнісна складність (розмір задач, які можна розв'язати таким алгоритмом)

Існують два базових поняття:

Поліноміальний алгоритм - алгоритм, функція оцінки якого зростає не швидше ніж поліном від n (де n - розмір)

Експоненціальний алгоритм - алгоритм, функція оцінки якого зростає швидше ніж поліном від n.

Щоб оцінити складність алгоритмів використовують вираз $O(f(n)) \to O$ - нотація.

Кажуть, що функція $f(n)$ порядку $g(n)$ для великих n:

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = const \neq 0.$$

Функція $f(n) \in o(g(n)) \to$ (o мале) для великих n:

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0.$$

- Якщо $f(n) \in O(g(n))$ - то дві ф-ї зростають однаково швидко
- Якщо $f(n) \in o(g(n))$ - то ф-я знаменника $g(n)$ зростає набагато повільніше

Деякі складності алгоритмів:

$O(1)$	Якщо час роботи не залежить від обсягу вхідних даних.
$O(n)$	Лінійна складність, яка прямо залежить від розміру. Тобто якщо подвоїти /потроїти вхідні дані, то подвоїться / потроїться і час виконання алгоритму відповідно.
$O(n^2)$	Квадратична складність, яка означає, що час роботи алгоритму зростає пропорційно квадрату кількості вхід даних. Якщо подвоїти розмір, то час збільшиться вчетверо.
$O(n^3)$	Кубічна складність, яка означає, що подвоєння розміру задачі збільшить необхідний час у 8 разів.
$O(\log n)$	Логарифмічна складність (бінарний пошук)

② Алгоритмічна система Тьюрінга. Машина Тьюрінга

Машина Тьюрінга - математична модель пристрою, який породжує обчислювальні процеси. Ця машина була названа на честь видатного математика Алана Тьюрінга. Це таке математичне поняття, яке дає формальне уточнення інтуїтивного поняття алгоритму.

Машина складається з таких основних компонентів:

- Головка для зчитування

Це такий елемент, що здатний читати, змінювати вміст комірок.

Головка здатна рухатись уздовж стрічки вправо (R)
вліво (L),
стояти на місці (N)

В кожен момент часу головка стоїть навпроти певної комірки і може виконувати вищенаведені операції з нею.

- інформаційна стрічка

Стрічка є безмежною в обидва боки λλ..... λλ...
Вона розбита на комірки, які можуть містити в собі лише інформацію про символ з алфавіту, який є фіксованим для кожної машини Тьюрінга (зовнішнього алфавіту)

Стрічка призначена для записування вхідної, проміжної, вихідної інформації.

- Пристрій керування.

Це такий функціональний механізм, який керує всією машиною. У кожен момент часу t пристрій керування перебуває у одному зі станів $Q = \{q_0, q_1 ... q_n\}$
Цю множину називають внутрішнім алфавітом.
Зазвичай початковим станом вважають стан q_0, а заключним станом - q_f, в якому машина припиняє свою роботу (інакше може відбутися зациклення)

Машину Тьюрінга використовують для розпізнавання мов найширшого

А мову, яку розпізнає МТ називають множиною слів із вхідних символів, допустимих

Для роботи МТ потрібно визначити множину команд (програму МТ) у вигляді:

$$< s_i, q_i > \rightarrow < s_j, q_r, Z >.$$

де s_i - символ з алфавіту

q_r - стан МТ

Z - напрямок руху голівки.

- Часова складність МТ → найбільша к-сть кроків, зроблених нею під час опрацювання команд.

Часова склад. не визначена, якщо на деякому вході n МТ не зупиняється.

- Ємнісна складність МТ → максимальна відстань, яку потрібно пройти голівці підчас опрацювання входу n

Машини Тьюрінга бувають:

- багатострічкові
- багатоголовкові
- з багатоповерховою стрічкою
- односторонні
- із декількома пристроями керування

Для прикладу, щоб краще пояснити як працює МТ наведу приклад машини, яка до будь-якого числа k в унарній системі з алфавітом A = {1,

застосує перетворення k + 2 в унарній сис-мі.

Отже, на початку машина стоїть зліва у стані q_0. У цьому стані вона шукає перший символ 1 і переходить у стан q_1, в якому рухається до кінця числа. Пройшовши усі "1" переходимо в стан q_2, який

замість порожнього символу поставить "1" і перейде у стан q_3, який додасть ще одну "1".
Якщо ж стан q_4 побачить на порожній символ, то робота успішно завершиться:

$\lambda\lambda\lambda 111 \rightarrow 11111\ \lambda\lambda$

	λ	1
q_0	$\lambda q_0 R$	$1 q_1 R$
q_1	$\lambda q_2 H$	$1 q_1 R$
q_2	$1 q_3 R$	
q_3	$1 q_4 R$	
q_4	$\lambda q_0 H$	

③ Показати елементарність ф-ї $f(x_1, x_2, x_3) = (x_1 + x_2) \cdot x_3$

1) $x_1 + x_2 = S(+, I_1^3, I_2^3)$
2) $(x_1 + x_2) \cdot x_3 = S(\cdot, S(+, I_1^3, I_2^3), I_3^3)$

Отже, маємо:
$f(x_1, x_2, x_3) = S(\cdot, S(+, I_1^3, I_2^3), I_3^3)$

Елементарну ф-ю можна отримати за допомогою оператора суперпозиції і найпростіших функцій:

1) Ф-я наступності: $S^1(x) = x + 1$
2) Тотожно рівна нулю $O(x) = 0$
3) Вибору аргументів: I_i^n.

Застосувавши такі функції я отримала, що ф-я є елементарною відносно + та *, отже, ф-я відповідно є елементарною.

ня

q_0 - перехід