Пошукові алгоритми

Пошук необхідної інформації – це одна з фундаментальних задач теоретичної інформатики та програмування. Саме пошукові алгоритми найчастіше є фрагментами різноманітних більш складних алгоритмів.

Пошуковими алгоритмами називають алгоритми, що здійснюють пошук потрібної інформації у множині заданих елементів.

Лінійний пошук

Найчастіше треба відшукати інформацію у невідсортованій послідовності елементів. У цьому випадку пошук зводиться до послідовного перегляду всіх елементів масиву, поки не знайдемо шуканий елемент. Такий пошук називається *лінійним*, або *послідовним*.

Алгоритм такого пошуку матиме такий вигляд:

1. Почнемо перегляд заданого масиву з першого елемента.
2. Поки шуканий елемент *x* не збігається з поточним елементом масиву *a[i]* і ми не вийшли за межі заданого масиву, то перейти до наступного елемента в масиві.
3. Якщо при відшуканні елемента *x* у масиві ми вийшли за його межі, тобто порядковий номер поточного елемента масиву сягнув значення *N*, то це означає, що шуканий елемент у заданому масиві відсутній. У протилежному випадку шуканий елемент знайдено на *i*-му місці в масиві.

Цьому фрагменту алгоритму відповідає такий текст Pascal-програми:

```
A[N+1]:=x;
i:=1;
while (a[i]<>x) do i:=i+1;
if i<N+1 then writeln('Шуканий елемент знайдено')
        else writeln('Шуканий елемент не знайдено');
```

Бінарний пошук

Пошук діленням навпіл

Пригадаємо, як ми здійснюємо пошук у словнику:

1. Відкриваємо словник у довільному місці й дивимося: якщо шукане слово знайдено, то пошук припиняємо, в іншому випадку переходимо до наступного пункту;
2. Якщо слово знаходиться попереду, то друга частина словника нас уже не цікавить і ми відкриваємо його десь усередині першої частини і починаємо пошук так само, як і в пункті 1);
3. Якщо слово знаходиться далі, то перша частина словника нас не цікавить і ми відкриваємо його десь усередині другої частини та починаємо пошук так само, як і в пункті 1).

Це є алгоритм пошуку інформації в упорядкованій послідовності елементів. Він носить назву бінарного (двійкового) пошуку, або пошуку методом поділу навпіл.

Визначимося щодо нашої задачі в алгоритмічних термінах. Нам задано масив a_i, $i=1,2,\ldots,n$, для елементів якого справджується умова: $a_i \le a_i+1$, $i=1,2,\ldots,n-1$. Шуканий елемент позначимо x, а елемент масиву, відносно якого масив ділиться на дві частини, – через $a[m]$.

Тепер алгоритм бінарного пошуку виглядатиме так:

1. Визначимо m.
2. Якщо $a[m]=x$, то пошук завершено і переходимо до п.5, в іншому випадку переходимо до п.3.
3. Якщо $a[m]<x$, то всі елементи з індексами, меншими за m, можна відкинути і перейти до п.1.
4. Якщо $a[m]>x$, то всі елементи з індексами, більшими за m, можна відкинути і перейти до п.1.
5. Вивести результат $a[m]$.

Найефективніший алгоритм бінарного пошуку заданого елемента в упорядкованій послідовності виглядатиме так:

```
L:=1; R:=N;
while (L<R) do
  begin
    m:=(L+R) div 2;
    if a[m] <x then L:=m+1 else R:=m
  end;
if a[R]=x then writeln('Шуканий елемент знайдено')
          else writeln('Шуканий елемент не знайдено');
```

Пошук у рядку

Прямий пошук підрядка у рядку

Нехай треба визначити можливість входження тексту *x*, що складається з *m* символів, у текст *s*, що складається з *n* символів *(m<n)*.

Будемо послідовно суміщати символи підрядка із символами рядка, починаючи з першого.

1	2	3		k		m	m+1		n-1	n
1	2	3		k		m				

При першому ж незбіганні деякого символу підрядка *x*, тобто при виконанні умови $x_k \neq s_k$ для $k \leq m$ (на малюнку зображено сірим кольором), зміщуємо накладання підрядка x на рядок s далі вправо на один символ: x_1 порівнюватимо з s_2, x_2 з s_3 і т.д.

1	2	3		k+1	m+1		n-1	n
	1	2		k	m			

Якщо ж і в цьому разі збігання з відповідним символом рядка *s* не відбудеться для деякого x_k, де $k \leq m$, то зміщення підрядка *x* по рядку *s* продовжимо далі.

Цей процес припиниться в одному з двох випадків:

- або на якомусь кроці символи всього підрядка *x* збігатимуться з фрагментом рядка *s*, і це означатиме, що ми знайшли входження підрядка *x* у рядок *s*;
- або нас весь час будуть супроводжувати не збігання символів підрядка *x* з символами рядка *s*, і це означатиме, що підрядок *x* жодного разу не входить до рядка *s*.

На мові Pascal:

```
n:=length(s);
m:=length(x);
i:=0;
flag:=false;
repeat
  inc(i);
  j:=1;
  while (x[j]=s[i+j-1]) and (j<=m) do inc(j);
  if j>m then flag:=true;
until flag or (i>n-m+1);
if flag then writeln('Позиція входження ',x,' у ',s,':',i)
        else writeln('Текст ',x,' не входить у текст ',s,' жодного разу');
```

Література:

Караванова Т.П. Інформатика: Методи побудови алгоритмів та їх аналіз. Необчисл. алгоритми: Навч. посіб. для 9-10 кл. із поглибл. вивч. інформатики. – К.:Генеза, 2007. – 216 с.: іл. – Бібліогр.: с. 212.