

Овсянников Олександр Сергійович
Бердянський державний
педагогічний інститут
ім. П.Д. Осипенко

Формування дослідницьких навичок студентів при розв'язанні задач порівняння й оптимізації алгоритмів

Мета курсу «Основ інформатики й обчислювальної техніки» - не тільки одержання студентами конкретних знань, умінь і навичок, але і розвиток їхніх спроможностей до розширення й удосконалення цих знань, умінь і навичок, розширення інтелектуальних можливостей студентів, формування дослідницьких навичок [1].

Дослідницькі навички застосовуються студентами на різних етапах процесу навчання у вищій школі - на етапі пошуку студентами нових знань, на етапі закріплення (дослідницька і, частково, пошукова діяльність студентів на семінарських, практичних, лабораторних заняттях), на етапі творчого застосування знань у процесі виконання курсових і дипломних робіт.

Розглянемо кілька задач, що можуть бути запропоновані студентам при вивченні основ програмування.

Задача №1.

Вам пропонується текст готової програми. Необхідно проаналізувати її, з'ясувати, що вона обчислює і розв'язати дану задачу іншим більш ефективним методом.

```
Program F;  
uses crt;  
var  
    n : integer;
```

```

f : longint;
i : integer;
begin
  clrscr;
  f := 1;
  write (' Введіть число --> ');
  readln (n);
  for i := 1 to n do
    begin
      f := f * i;
    end;
  writeln ('Число ',n,' після деяких перетворень дорівнює ',f);
  readln;
end.

```

Одним із запропонованих студентами рішень може бути така програма.

```

Program Factorial_Recurcia;
Uses crt;
Var
  n : integer; {число факторіал якого треба обчислити}
  f : longint; {факторіал числа n}
function factorial(k : integer) : longint;
  begin
    if k = 1
      then factorial := 1
      else factorial := k * factorial(k - 1);
    end;
begin {основна програма}
  clrscr;

```

```
writeln ('Обчислення факторіала з використанням рекурсивної функції');  
write ('Введіть число, факторіал якого треба обчислити --> ');  
readln (n);  
f := factorial(n);  
writeln ('Факторіал числа ',n,' дорівнює ',f);  
readln;  
end.
```

Задача № 2.

Вам пропонується алгоритм і готовий текст програми **Masiv1**, яка виконує впорядкування масиву цілочисельних даних. Потрібно оптимізувати алгоритм її роботи або написати інший варіант розв'язання цієї задачі.

Алгоритм сортування масиву за зростанням методом прямого вибору може бути поданий так:

1. Переглядаючи масив від першого елемента, знайти мінімальний і помістити його на місце першого елемента, а перший на місце мінімального.
2. Переглядаючи масив від другого елемента, знайти мінімальний і помістити його на місце другого елемента, а другий на місце мінімального.
3. Так продовжувати до останнього елемента.

```
Program Masiv1;  
{впорядкування масиву методом прямого вибору}  
Uses crt;  
Const  
    size = 5;  
Var  
    a : array[1..size] of integer;
```

i : integer; {номер елемента, від якого ведеться пошук мінімального елемента
}

min : integer; {номер мінімального елемента в частині масиву від i
до верхньої межі масиву}

j : integer; {номер елемента, порівнюваного з мінімальним}

buf : integer; {буфер, використовуваний при обміні елементів масиву}

k : integer;

begin

clrscr;

writeln (' Впорядкування масива');

write ('Введіть',size:3,' цілих чисел ');

writeln (' і натисніть <Enter>');

for k:=1 to size do

begin

readln (a[k]);

end;

writeln(' впорядкування ');

for i:= 1 to size-1 do

begin

{ пошук мінімального елемента в частині масиву від a[i] до
a[size] }

min:=i;

for j:=i+1 to size do begin

if a[j]<a[min] then min:=j;

end;

begin

{ поміняємо місцями a[min] і a[j] }

buf:=a[i];

a[i]:=a[min];

a[min]:=buf;

```
    {виведемо масив}  
    for k:=1 to size do write(a[k], ' ');  
    writeln;  
    end;  
end;  
writeln('Масив впорядкований. ');  
readln;  
end.
```

У основі алгоритму запропонованого студентами рішення програми сортування масиву цілочисельних даних за зростанням може лежати обмін сусідніх елементів масиву. Кожний елемент масиву, починаючи з першого, порівнюється зі слідуючим, і якщо він більший слідуючого, то елементи обмінюються місцями.

```
Program mas2;  
uses crt;  
const size=10;  
Var  
    a : array[1..size] of integer;  
    i : integer;  
    k : integer;  
    buf : integer;  
begin  
    clrscr;  
    writeln (' Впорядкування масиву. ');  
    writeln ('Введіть',size:3,' цілих чисел ');  
    for k:=1 to size do  
        begin  
            readln (a[k]);
```

```

    end;
writeln(' впорядкування ');
for i:= 1 to size-1 do
begin
    for k:=1 to size-1 do begin
        if a[k]>a[k+1] then
            begin
                {міняємо містами k-й і (k+1)-й елементи}
                buf:=a[k];
                a[k]:=a[k+1];
                a[k+1]:=buf;
            end;
        end;
    for k:=1 to size do write(a[k], ' ');
    writeln;
end;
writeln('Масив впорядкований. ');
readln;
end.

```

Задача № 3

Вам пропонуються тексти програм **Poisk** і **Bin_Poisk** пошуку елемента у впорядкованому масиві даних. Потрібно зазначити оптимальний варіант розв'язання задачі.

```

program Poisk;
uses
    Crt;
var
    massiv: array [1.. 10] of integer;

```

```

zrazok: integer;
znaideno:boolean;
i:integer;
begin
clrscr;
writeln('Пошук у масиві. ');
write ('Введіть 10 цілих в одному рядку ');
writeln (' і натисніть <Enter>');
write('->');
for i:=1 to 10 do read(massiv[i]);
write ('Введіть зразок для пошуку (ціле число)-> ');
readln(zrazok);
znaideno:=FALSE;
i:=1;
repeat
if massiv[i] = zrazok
then znaideno :=TRUE
else i:=i+1;
until (i>10) or (naiden) ;
if znaideno
then writeln (' Співпадання з елементом номер ', i:3, ' ',
'пошук успішний. ')
else writeln (' Співпадань із зразком немає. ');
readln;
end.

program Bin_Poisk;
uses
  Crt;
var

```

```
a: array [1..9] of integer;
zrazok: integer;
sred,verh,niz:integer;
znaideno:boolean;
n:integer;
i:integer;
begin
clrscr;
writeln ('Бінарний пошук у масиві. ');
write('Введіть 9 цілих в одному рядку ');
writeln(' і натисніть <Enter>');
for i:=1 to 9 do read(a[i]);
writeln ('Введіть зразок для пошуку (ціле число) ');
readln(zrazok);
verh:=1;
niz:=9;
znaideno:=FALSE;
n:=0;
writeln(' verh niz sred ');
repeat
sred:=(niz-verh) div 2 +verh;
writeln(verh:6,niz:6,sred:6) ;
n:=n+1;
if a[sred]= zrazok then znaideno:=TRUE
else begin
if zrazok <a[sred]
then niz:=sred-1
else verh:=sred+1;
end;
until (verh>niz) or znaideno;
```



```
if znaideno
then write ('Співпадання з елементом номер ',
sred, ' . Виконано ',n, ' порівнянь. ')
else writeln('Зразок у масиві не знайдений. ');
readln;
end.
```

Студенти повинні побачити що, у першому варіанті розв'язання задачі пошук здійснюється послідовним порівнянням елементів масиву зі зразком доти, поки не буде знайдений елемент, що співпадає зі зразком, або не будуть перевірені всі елементи.

Суть другого методу полягає в тому, що вибирається середній елемент упорядкованого за зростанням масиву (елемент із номером sred), з яких зрівнюється зразок.

Якщо середній елемент співпадає зі зразком, то задача розв'язана.

Якщо середній елемент менший ніж зразок, то шуканий елемент розташований вище середнього елемента (між елементами з номерами verh і sred).

Якщо середній елемент більше зразка, то шуканий елемент розташований нижче середнього (між елементами з номерами sred і niz)

Змінна n (підраховує кількість порівнянь із зразком) дозволяє оцінити ефективність другого алгоритму в порівнянні з пошуком методом простого перебору (для масивів упорядкованих за зростанням).

Задачі на оптимізацію алгоритмів і на порівняння ефективності алгоритмів відносяться до класу творчих задач і безумовно сприяють формуванню дослідницьких навичок студентів.

Література.

1. Чепрасова Т. І. Підвищення значущості результатів навчання інформатики в старших класах середньої школи в умовах НІТН. Дис. . . канд. пед. наук. 13.00.02. НПУ ім. Драгоманова. - К.: 1998.