Összegzés tétele:

```
algoritmus összegzés;
  összeg:=Ø;
  ciklus i:=1..n ismétel
      összeg:= összeg + A[i];
  c vége;
```

Megszámlálás tétele:

```
algoritmus megszámlálás;
  Db:=Ø;
  ciklus i:=1..n ismétel
    ha A[i] P tulajdonságú akkor
        Db:=Db+1;
    h vége;
  c vége;
```

Kiválogatás tétele:

```
algoritmus kiválogatás;
  Db:=Ø;
  ciklus i:=1..n ismétel
    ha A[i] P tulajdonságú akkor
      Db:=Db+1;
      AP[Db]:=A[i];
    h vége;
  c vége;
```

Kiválasztás tétele:

algoritmus kiválasztás;

i:= 1;

amíg A[i] nem P tulajdonságú ismétel

i:= i+1;

a vége;

hely:=i;

Eldöntés tétele:

algoritmus eldöntés;

i:=1;

amíg (i<=n) és (A[i] nem P tulajdonságú) ismétel

i:=i+1;

a vége;

van:=(i<=n);

Lineáris keresés:

algoritmus lineáris keresés;

i:= 1;

amíg (i<=n) és (A[i] nem P tulajdonságú) ismétel

i:=i+1;

a vége;

hely:=i;

van:= (i<=n);

Minimumkiválasztás tétele:

```
algoritmus minimumkiválasztás;
  min := 1;
  ciklus i:=2..n ismétel
    ha A[i] < A[min] akkor
      min := i;
    h vége;
  c vége;
```

Minimumkiválasztásos rendezés:

```
algoritmus minimumkiválasztásosrendezés;
  ciklus j:=1..n-1 ismétel
    min :=j;
      ciklus i:= j+1..n ismétel
        ha A[i]<A[min] akkor
          min := i;
        h vége;
      c vége;
      csere:= A[min];
      A[min] := A[j];
      A[j] := csere;
  c vége;
```

Keresések deklarációja:

konstans n = maximális_elemszám

típus elemtípus : tárolandó_elemek_típusa

típus tömbtípus : tömb [1..n] elemtípus

Lineáris keresés:

```
függvény lineáris_keresés (A: tömbtípus; Adat : elemtípus;
                                   hely : egész ): logikai;
  változó  i: egész ;
  i := 1;
  amíg (i<=n) és (A[i]<>Adat) ismétel
     i:=i+1;
  a vége ;
  hely :=i;
  lineáris_keresés := (i<=n) ;
vége;
```

Keresés rendezett sorozaton:

```
függvény rendkeresés ( A: tömbtípus; Adat: elemtípus;
                                   hely: egész): logikai;
  változó  i:egész;
  i:=1;
  amíg (i<=n) és (A[i]< Adat) ismétel
     i:=i+1;
  a vége;
  hely := i;
  rendkeresés := (i<=n) és (A[i] = Adat);
vége;
```

Strázsás keresés:

függvény strázsáskeresés (A: tömbtípus ; Adat : elemtípus ; hely : egész) : logikai ;

változó i: egész;

A[n+1] := Adat;

i := 1;

amíg A[i] <> Adat ismétel

i := i+1;

a vége;

hely := i;

strázsáskeresés := (i<=n) ;

vége;

lineáris keresés:

ciklus végrehajtás: min: 1 ; max: n ; átlag: $\frac{n+1}{2}$

összehasonlítás : n után nincs meg a keresett elem

keresés rendezett sorozaton:

ciklus végrehajtás: min: max: átlag: $\frac{n+1}{2}$

összehasonlítás : $\frac{n+1}{2}$ után nincs meg a keresett elem

strázsás keresés:

ciklus végrehajtás: min: 1 ; max: n ; átlag: $\frac{n+1}{2}$

összehasonlítás : n+1 után nincs meg a keresett elem

Bináris keresés:

```
függvény binráriskeresés (A: tömbtípus; Adat: elemtípus; hely: egész):
                        logikai;
 változó e, u, k: egész;
   e:= 1;
   u:= n;
   k:= (e+u)/2;
   amíg (e<=u) és (A[k]<>Adat) ismétel
     elágazás
       amikor Adat < A[k]:
          u:= k-1;
       amikor Adat > A[k]:
          e:= k+1;
     e vége;
     k:= (e+u)/2;
   a vége;
   hely := k;
   bináriskeresés := e<=u;
vége;
```

összehasonlítások száma:

min: 1 max: $\log_2(n)+1$ átl: $\log_2(n)$

Minimumkiválasztásos rendezés:

```
eljárás minimumkivrend (A : tömbtípus);
  változó i,j, min : egész;
  változó minérték : elemtípus;
  ciklus i := 1 .. n-1 ismétel
    min := i;
    ciklus j := i+1 .. n ismétel
      ha A[min] > A[j] akkor
        min := j;
      h vége;
    c vége;
    ha min <> i akkor
      minérték := A[min];
      A[min] := A[i];
      A[i] := minérték;
    h vége;
  c vége;
vége;
```

hatékonyság:

helyfoglalás: $n+1$

összehasonlítások száma: $\frac{n \cdot (n-1)}{2}$

értékadások száma: rendezett sorozat esetén : 0

rendezetlen sorozat esetén : $3(n-1)$

Buborékrendezés:

```
eljárás buborékrend (A: tömbtípus);
  változó i, j: egész;
  változó csere: elemtípus;
  ciklus i := n .. 2 lépésköz -1 ismétel
    ciklus j := 1 .. i-1 ismétel
      ha A[j] > A[j+1] akkor
        csere := A[j];
        A[j] := A[j+1];
        A[j+1] := csere;
      h vége;
    c vége;
  c vége;
vége;
```

hatékonyság:

helyfoglalás: $n+1$

összehasonlítások száma: $\frac{n \cdot (n-1)}{2}$

értékadások száma: rendezett sorozat esetén: 0

fordított rendezettségnél: $\frac{3 \cdot n \cdot (n-1)}{2}$

Javított buborékrendezés I.

```
eljárás javítottbuborékrendezés1 (A: tömbtípus);
  változó i, j: egész;
  változó csere: elemtípus;
  változó vége: logikai;
  i: = n;
  vége := hamis;
  amíg (i>= 2) és (nem vége) ismétel
    vége: = igaz;
    ciklus j: = 1 .. i-1 ismétel
      ha A[j] > A[j+1] akkor
        csere: = A[j];
        A[j]: = A[j+1];
        A[j+1]: = csere;
        vége: = hamis;
      h vége;
    c vége;
    i: = i-1;
  c vége;
vége;
```

Hatékonyság:

Helyfoglalás: $n+1$

Összehasonlítások száma: eleve rendezett sorozat: $n-1$

fordított rendezettségnél: $n \cdot (n-1)/2$

Értékadások száma: eleve rendezett sorozatnál: 0

fordított rendezettségnél: $3 \cdot n \cdot (n-1)/2$