

```

function SELECT(A,i)
    // find i-th element in array A
    p:=choose_pivot(A);
    //--- partition A into LESS, EQUAL, MORE
    create new arrays LESS, EQUAL, MORE;
    for i:=1 to size(A) do
        if A[i]<p then add A[i] to LESS;
        if A[i]=p then add A[i] to EQUAL;
        if A[i]>p then add A[i] to MORE;
    //--- decide, what case to pursue
    if size(LESS)>=i then
        return SELECT(LESS,i);
    else if size(LESS)+size(EQUAL)>=i then
        return p;
    else
        return SELECT(MORE,i-size(LESS)-size(EQUAL));

```

```
function SELECT(A,i)
* if size(A)<100 then
*   sort elements of A;
*   return A[i];
else
  // find i-th element in array A
  p:=choose_pivot(A);

  // partition A into LESS, EQUAL, MORE
  create new arrays LESS, EQUAL, MORE;
  for i:=1 to size(A) do
    if A[i]<p then add A[i] to LESS;
    if A[i]=p then add A[i] to EQUAL;
    if A[i]>p then add A[i] to MORE;

  ....
```

## Časová zložitosť deterministického algoritmu

Časová zložitosť algoritmu  $A$  je **funkciou veľkosti vstupu**, pričom  $T_A(n)$  je najväčší čas potrebný na vyriešenie vstupu o veľkosti  $n$ .

$$T_A(n) = \max\{T_A(x) \mid |x| = n\}$$

( $T_A(x)$  je čas, ktorý algoritmus  $A$  potrebuje na vyriešenie vsupu  $n$ .)

## Časová zložitosť pravdepodobnostného algoritmu

Časová zložitosť algoritmu  $A$  je **funkciou veľkosti vstupu**, pričom  $T_A(n)$  je najväčší čas potrebný na vyriešenie vstupu o veľkosti  $n$ .

$$T_A(n) = \max\{E_R[T_A(x, R)] \mid |x| = n\}$$

( $T_A(x, R)$  je čas, ktorý algoritmus  $A$  potrebuje na vyriešenie vstupu  $n$  za predpokladu, že postupnosť príslušných náhodných rozhodnutí je  $R$ .)