Sortowanie bąbelkowe

Pseudokod sortowania tablicy jednowymiarowej

```
procedure SortBabelStd
var
    i,j : indeks;
    x  : obiekt;

begin
  for i:=2 to n do
    for j:=n downto i do
      if a[j-1]>a[j] then
        begin
          x:=a[j-1];
          a[j-1]:=a[j];
          a[j]:=x;
        end;
end;
```

Pseudokod sortowania jednowymiarowej tablicy rekordów

```
procedure SortBabelStd
var
    i,j : indeks;
    x  : obiekt;

begin
  for i:=2 to n do
    for j:=n downto i do
      if a[j-1].klucz>a[j].klucz then
        begin
          x:=a[j-1];
          a[j-1]:=a[j];
          a[j]:=x;
        end;
end;
```

Zoptymalizowane sortowanie bąbelkowe:

1. Można zapamiętać w zmiennej logicznej, czy w ostatnim kroku były jakieś zamiany. Jeśli nie, to algorytm można zakończyć.
2. Można zapamiętać indeks k, dla którego w ostatnim kroku dokonano ostatniej zamiany. Oznacza to, że poniżej elementu k-tego wszystkie następne są posortowane.

```
procedure SorBabelOpt
var
    i,j,k :   indeks;
    x     :   obiekt;
    BylaZam: Boolean;

begin
   i:=2;
   k=i;
   BylaZam:=TRUE;
   while ((i<=n) And BylaZam) do
       begin

          BylaZam:=FALSE;    {jeśli będzie, to ją przestawimy}
          for j:=n  downto k do
            if a[j-1].klucz>a[j].klucz then
               begin
                   BylaZam:=TRUE;
                   k:=j;
                   x:=a[j-1];
                   a[j-1]:=a[j];
                   a[j]:=x;
               end      {if}

       end;     {while}
end;
```