Třídění, rozdílové seznamy

bubblesort(S,Sorted)

Seznam S setřiďte tak, že

- nalezněte první dva sousední prvky X a Y v S tak, že X>Y,
  vyměňte pořadí X a Y a získate S1;
  a setřiďte S1
- pokud neexistuje žádný takový pár sousedních prvků X a Y,
  pak je S setříděný seznam

```
swap([X,Y|Rest],[Y,X|Rest1]) :-
        X>Y.                         % nebo obecněji X@>Y pomocí gt(X,Y)
swap([Z|Rest],[Z|Rest1]) :-
        swap(Rest,Rest1).

bubblesort(S,Sorted) :-
        swap (S,S1), !,
        bubblesort(S1, Sorted).
bubblesort(Sorted,Sorted).
```

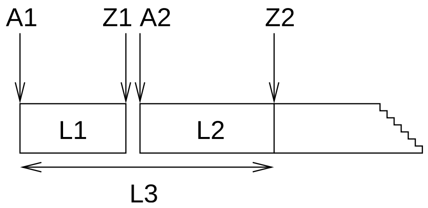

quicksort(S,Sorted)

Neprázdný seznam S setřiďte tak, že

- smažte nějaký prvek X z S;
  rozdělte zbytek S na dva seznamy Small a Big tak, že:
  v Big jsou větší prvky než X a v Small jsou zbývající prvky
- setřiďte Small do SortedSmall
- setřiďte Big do SortedBig
- setříděný seznam vznikne spojením SortedSmall a [X|SortedBig]

```
quicksort([],[]).                      + ošetření případu, kdy S je prázdný seznam
quicksort([X|T], Sorted) :- split(X, Tail, Small, Big),
                            quicksort(Small, SortedSmall),
                            quicksort(Big, SortedBig),
                            append(SortedSmall, [X|SortedBig], Sorted).
split(X, [], [], []).
split(X, [Y|T], [Y|Small], Big) :- X>Y, !, split(X, T, Small, Big).
split(X, [Y|T], Small, [Y|Big]) :- split(X, T, Small, Big).
```


Rozdílové seznamy

- Zapamatování konce a připojení na konec: rozdílové seznamy
- [a,b] = L1-L2 = [a,b|T]-T = [a,b,c|S]-[c|S] = [a,b,c]-[c]
- Reprezentace prázdného seznamu: L-L
- 

- ?- append([1,2,3|Z1]-Z1, [4,5|Z2]-Z2, S).
- append(A1-Z1, Z1-Z2, A1-Z2).
  L1 L2 L3

reverse(Seznam, Opacny)

```
reverse( [], [] ).
reverse( [ H | T ], Opacny ) :-
        reverse( T, OpacnyT ),
        append( OpacnyT, [ H ], Opacny ).
```

```
reverse( Seznam, Opacny ) :- reverse0( Seznam, Opacny-[]).
reverse0( [], S-S ).
reverse0( [ H | T ], Opacny-OpacnyKonec  ) :-
         reverse0( T,  Opacny-[ H | OpacnyKonec]  ).
```

```
reverse( Seznam, Opacny ) :- reverse0( Seznam, [], Opacny ).
reverse0( [], S, S ).
reverse0( [ H | T ], A, Opacny ) :-
         reverse0( T, [ H | A ], Opacny ).
```


quicksort **pomocí rozdílových seznamů**

Neprázdný seznam S setřiďte tak, že

- smažte nějaký prvek X z S;
  rozdělte zbytek S na dva seznamy Small a Big tak, že:
  v Big jsou větší prvky než X a v Small jsou zbývající prvky
- setřiďte Small do SortedSmall
- setřiďte Big do SortedBig
- setříděný seznam vznikne spojením SortedSmall a [X|SortedBig]

```
quicksort(S, Sorted) :- quicksort1(S,Sorted-[]).

quicksort([],Z-Z).
quicksort([X|T], A1-Z2) :-
        split(X, Tail, Small, Big),
        quicksort(Small,  A1-[X|A2]),
        quicksort(Big,  A2-Z2).                append(A1-Z1, Z1-Z2, A1-Z2).
```