

Merge sort—hlavný program

```
// sort sequence A[l..r]
function merge_sort(l,r)
    // base case - 1 element is always sorted
    if (l=r) then return;
    m=(l+r) div 2;
    // we need to sort sequences l..m, m+1..r
    merge_sort(l,m);
    merge_sort(m+1,r);
    // and finally merge two sorted sequences
    merge(l,m,r);
```

Merge sort—merge

```
//merge two sorted sequences l..m, m+1..r
function merge(l,m,r)
  copy A[l..m] to L; L[m-1+2]:=infinity;
  copy A[m+1..r] to R; R[r-m+1]:=infinity;

  i:=1; j:=1; k:=1;
  while (L[i]<infinity or R[i]<infinity) do
    if L[i]<=R[j] then
      A[k]:=L[i];
      i:=i+1; k:=k+1;
    else
      A[k]:=R[j];
      j:=j+1; k:=k+1;
```

Rozdeľuj a panuj

- **Rozdeľuj.** Rozdeľ problém na niekoľko menších podproblémov.
- **Panuj.** Každý podproblém vyrieš samostatne rekurzívnym volaním. Ak sú podproblémy dostatočne malé, vyrieš ich priamočiaro.
- **Kombinuj.** Skombinuj riešenia menších podproblémov do riešenia pôvodného veľkého problému.

Hlavná veta (master theorem): Nech $T(n) = aT(n/b) + f(n)$,
 $T(1) = \Theta(1)$. Nech $k = \log_b a$. Potom:

1. Ak $f(n) \in O(n^{k-\varepsilon})$ pre niektoré $\varepsilon > 0$, potom $T(n) \in \Theta(n^k)$.
2. Ak $f(n) \in \Theta(n^k)$, potom $T(n) \in \Theta(f(n) \log n)$.
3. Ak $f(n) \in \Omega(n^{k+\varepsilon})$ pre niektoré $\varepsilon > 0$ a platí podmienka regularity, potom $T(n) \in \Theta(f(n))$.

Podmienka regularity: Existuje $c < 1$ také, že pre všetky dostatočne veľké n platí $af(n/b) \leq cf(n)$.

Poznámka: Veta platí aj v prípade rozumných usporiadaní dolných a horných celých častí - vid' napr. CLRS2 4.4.2.

