Data je jednostruko povezana lista prirodnih brojeva. Algoritamski rešiti problem nalaženja maksimalnog elementa liste (vrednost + pokazivač na čvor).

```
// Ulaz: lista l prirodnih brojeva
// Izlaz: maksimalna vrednost u listi i pokazivač na čvor
//        koji sadrži maksimalnu vrednost

algorithm maxp(l)

max = -1;
pmax = null;
p = l;

while (p != null) do

    if (p.kljuc > max) then      //
        max = p.kljuc;
        pmax = p;

    p = p.sled;                  // sledeći čvor


return max,pmax;       // vrati maks.vrednost i pmax
```

Data je jednostruko povezana lista sortirana u rastući poredak. Algoritamski rešiti problem dodavanja elementa x tako da lista ostane sortirana.

```
// Ulaz: Sortirana lista l i čvor q koji sadrži vrednost x
// Izlaz: Lista l sa dodatim čvorom q(odnosno elementom x)

algorithm sort_insert(l,q)

if (l == null) then
    list-insert(q,null,l);

if (l.kljuc >= q.kljuc) then
    list-insert(q,null,l);

p = l;    // inicijalizujemo pomoćni pokazivač na glavu

while ((p.sled != null)&& (p.sled.kljuc < q.kljuc)) do

    p = p.sled;                    // sledeći čvor

list-insert(q,p,l);
```

Data je dvostruko povezana lista. Algoritamski rešiti problem prebacivanja repa liste na poziciju glave liste.

```
// Ulaz: Dvostruko povezana lista l
// Izlaz: Lista l sa promenjenom pozicijom repa liste -
//        rep liste se premešta na poziciju glave liste

algorithm rep_glava(l)

if (l == null) then
    return 0;   // lista je prazna

if (l.sled == null) then
    return 1;   // lista ima jedan čvor glava=rep!

p = l;    // inicijalizujemo pomoćni pokazivač na glavu

while ((p.sled != null) do

       p = p.sled;                  // sledeći čvor

p.pret.sled = null;

p.pret = null;

p.sled = l;

l.pret = p;

l = p;
```