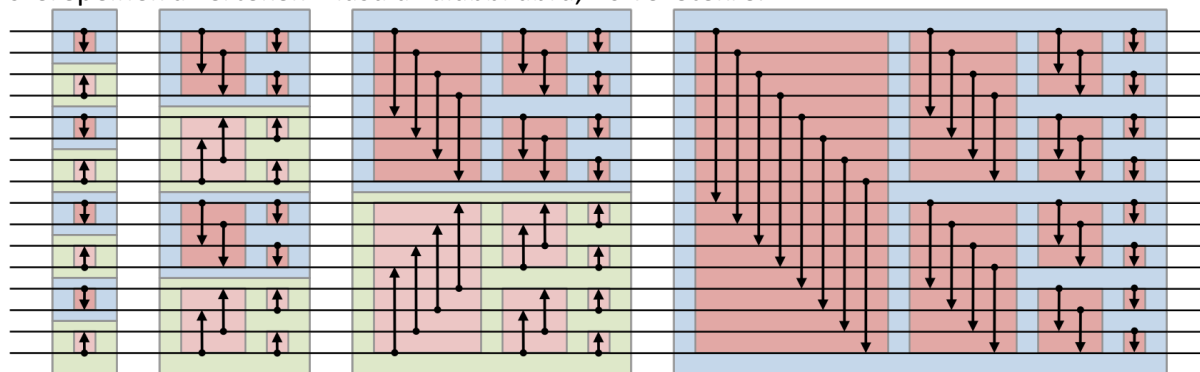


## 4. Házi Feladat

### Rendező hálózat

A feladat egy rendező hálózat implementációja: a hálózat bemenete egy  $n$  hosszú rendezetlen számsorozat elemei, kimenete pedig a rendezett sorozat. A hálózat elképzelhető, mint egy áramkör,  $n$  vízszintes vezetékkel: az áramkör elején az  $n$  vezetékre bal oldalon rákapcsoljuk a sorozat értékeit, a végén pedig az  $n$  vezeték kimenetén jobb oldalon már sorrendben szerepelnek az értékek – lásd az alábbi ábra, 16 vezetékre.



A nyilak összehasonlító és cserélő műveleteket jeleznek: amikor két szám eléri a nyíl két végét, összehasonlítjuk őket és ha kell megcseréljük őket, hogy a nyíl mindig a *nagyobbik* felé mutasson.

Készíts el egy algoritmust, melynek bemenete egy számsorozat egy tömbben (mérete 2 hatványa), az  $n=2^k$  vezetéknek megfelelően, a futást követően ugyanez a tömb az eredményt növekvő sorrendben tartalmazza. A kód működjön tetszőleges 2 hatványra.

A hálózat blokkokból áll (kék és zöld téglalapok) – minden blokk pedig  $k$  lépésből (piros téglalapok), ahol  $2^k=n$ ,  $n$  pedig a blokk bemeneteinek száma. Egy blokk vagy növekvő, vagy csökkenő sorrendbe rendez (kék növekvő, zöld csökkenő). Egy blokk lépései rendre  $n/2$ ,  $n/4$ , stb. lépésközzel végzik el az összehasonlítást és cserét. A hálózatban visszafelé haladva látható, hogy minden egyes blokk bemenete két másik blokk kimenete – melyekből az első mindig növekvő (kék), a második pedig csökkenő (zöld) sorrendbe rendez. A hálózatot felépíteni így visszafelé haladva célszerű, a blokkon belüli lépések számát mindig csökkentve, illetve a blokkok számát mindig duplázva, a megfelelő csökkenő/növekvő rendezési iránnyal. Rekurzív implementáció javasolt.

A rendező algoritmust a következő szignatúrájú függvénnyel lehessen meghívni, az eredmény a függvény lefutása után a bemeneti tömbben legyen:

```
template <typename T>
void rendezo_halozat(T *tomb, int meret)
```