**Working in parallel**

If *i objects* working in parallel, the **result** is always written like:

R = 1/a + 1/b + …. + 1/i

The **result** is **1/R**

Example:

* How much, N taps filling a tank **at the same time** ?
* How much time, for painters that paint **at same time**

**MIN , MAX and The index of a value**

To return the **min** of an array:

min\_of\_arr = **\*min\_element(** arr , arr+n )

To return the **max** of an array:

min\_of\_arr = **\*max\_element(**arr , arr+n)

to return **the index** of a value in the array : **distance( iter0 , iter1)**

index\_of\_min = **distance(** arr , *min\_element(* arr , arr +n *)*  **)**

**IMPORTANT:**

* *distance()* on accepts **ITERATORS** not values !
* *min\_element()* **DOES NOT** return values , so we should **ADD \***