~~while~~

Iterative while

2) Runtime

1) Correctness

Runtime

```
i=0
while( i < n )  { // n iterations
    ++i;
    if( a[i] < a[j] ) {SWAP}   <- "≤"
    printf( .... );
}
```

- $O(n)$ worst
- $O(\frac{n}{2})$ average (10^6)
- $O(1)$ best

RT is $O(n)$

$O(n)$ foo.cpp $\xrightarrow[\text{email}]{\text{~~sen~~}}$ foo.cpp $(O(n))$

1 s. ~~1 s~~

n = 1000 run foo 1 s

↓ ↓

n = 10 000 — 10 s w/o run.

$O(n^3) = k n^3$

$n = 10^3$ — 3 sec. $k(10^3)^3 = 3$ s.

$n = 10^6$ — ? 5 month. $k(10^6)^3 =$

$= k \cdot 10^{18} = (k \cdot 10^9) 10^9$ ($k \cdot 10^9$ = 3 s.)

$= 3 \cdot 10^9$ sec.

$= 3~~\#~~31.7$ years.

```
for (int i=0; i<n; ++i) {          //n
    for( int j=0; j<n; ++j {       //n   } O(n^2)
    }
}
```

```
for( i=0; i<n; ++i)
    for(j=0; j<i; ++j)
```

i=0 1 2

$0+1+2+3+\ldots+n-1 =$

$= \frac{a_1+a_n}{2} \cdot \frac{n}{2} = \frac{0+n-1}{2} \cdot \frac{n}{2} =$

$= \frac{n(n-1)}{2} = O(n^2)$

```
for(i=0; i<n; ++i)                 //n
    for( j=0; j<n; j=j+10)         // n/10   } O(n^2)
```

Binary search. — slides.

$$1 + 2 + 4 + \cdots + 2^{F-1} = \frac{2^F - 1}{2-1} = 2^F - 1$$

$O(n^2)$ "$\leq$"

$\Theta(n^2)$ "$=$"

---

while ($i = \frac{n}{2}$; $i \geq 0$; ++i) { // $n/2$

"$\leq \log_2 n$"

3

$O(n \log n)$

$O(n)$

$n = 10^6 \approx (1024)^2 = (2^{10})^2 = 2^{20}$

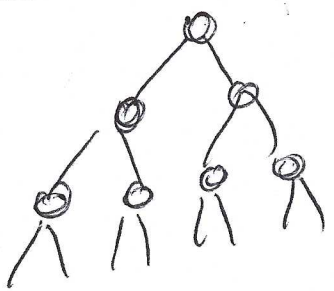
$\log n$ $\quad \log_2 10^6 \approx 20$

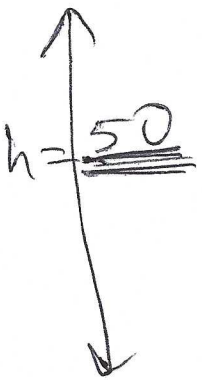
n $\quad \approx 1000000$ ↘ 50 000 times.

$\sqrt{n}$ $\quad \approx 1000$ ↘ 50 slower.

```
foo( ... )
{
    base
    l = foo( ... )
    r = foo( ... )
    if (l<r) return 1;
    else return 10;
}
```

1ms





$1 + 2 + 4 + 8 + \dots + 2^{50}$

$2^{51} - 1$

$O(2^n)$

$2^{50} \cdot \frac{1}{1000} = 2^{47}$ sec. = 1,628,906,480 year

1.5B years.

atoms in universe $\approx 2^{240}$

Correctness

Code

$x = x+2$ // ~~$=$~~ $x := x+2$

$x \leftarrow x+2$

$\underline{\underline{x}}$ variable

Math.

$0 = 2$

~~$x = x+2$~~ has no solution

$\underline{\underline{x}}$

$\forall x \; \underbrace{x^2 \geq 0}$

x - constant

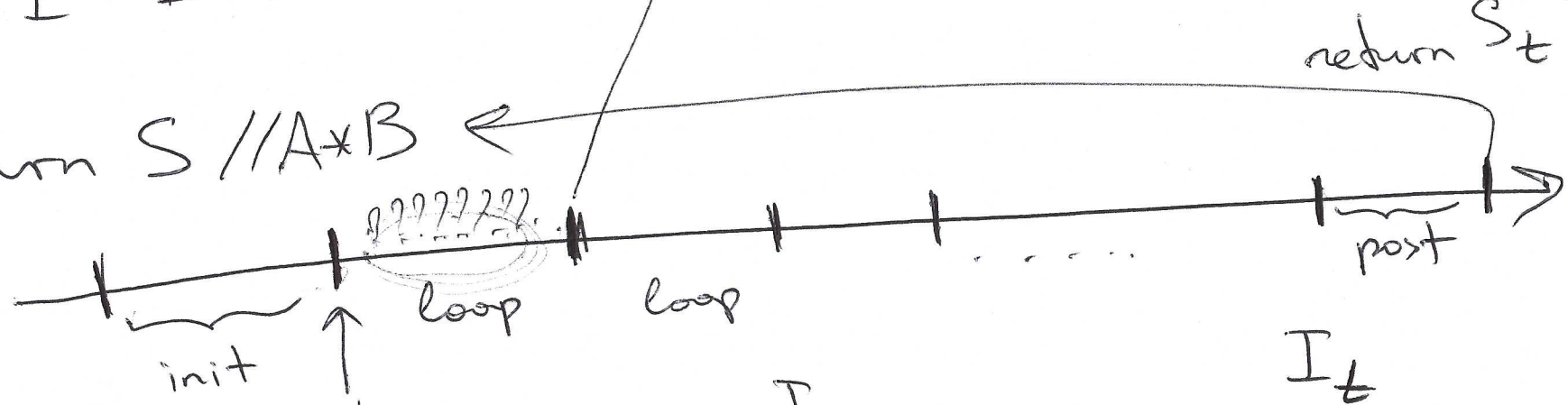
$x^2 = x + 1$

$x_{1,2} = \frac{1 \pm \sqrt{5}}{2}$

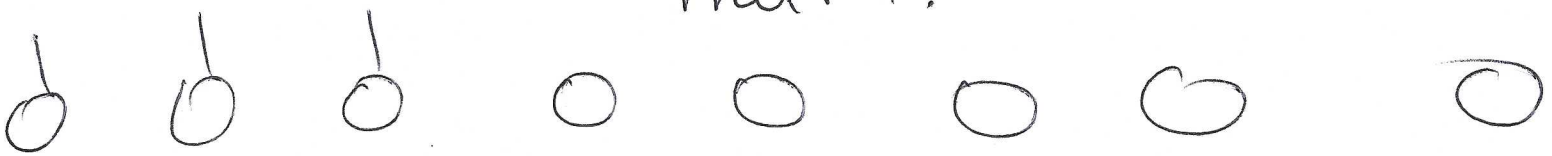
```
Alg (A, B) { // B - int , B>0
             (if (B<0) return NULL;)
  S = A
  I = 0
  while (I < B) {
      S = S + A
      I = I + 1
  }
  return S //A*B
}
```

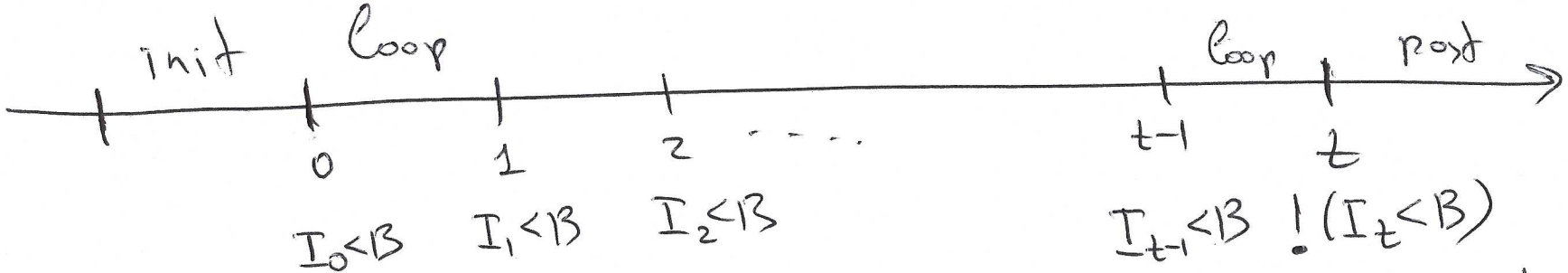
C++ A, B are constan

use in math.



math.





Invariant: math. statement about variables that is always true

"main idea"

$1A, 2A, 3A \ldots$

$S_1 \; S_2 \; S_3$

$\begin{cases} S_k = k \cdot A \\ I_k = k \end{cases}$? proof.