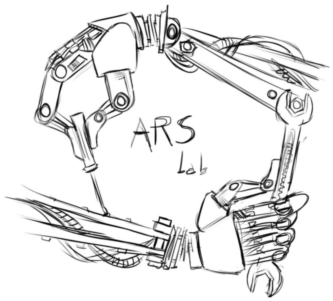
Bubble Sort e Prodotto Scalare in Assembly

Corrado Santoro

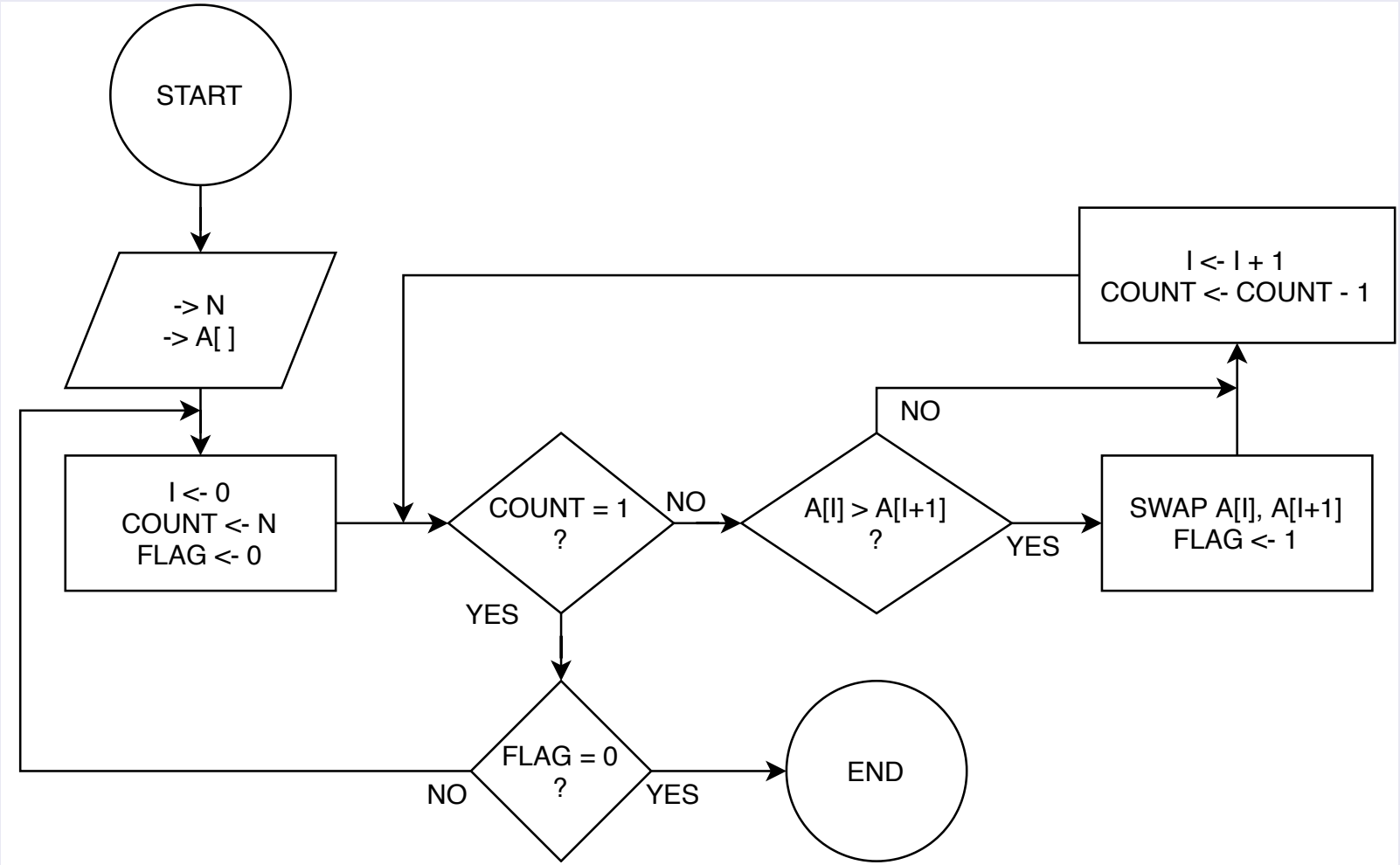
Dipartimento di Matematica e Informatica
santoro@dmi.unict.it



Corso di Architettura degli Elaboratori

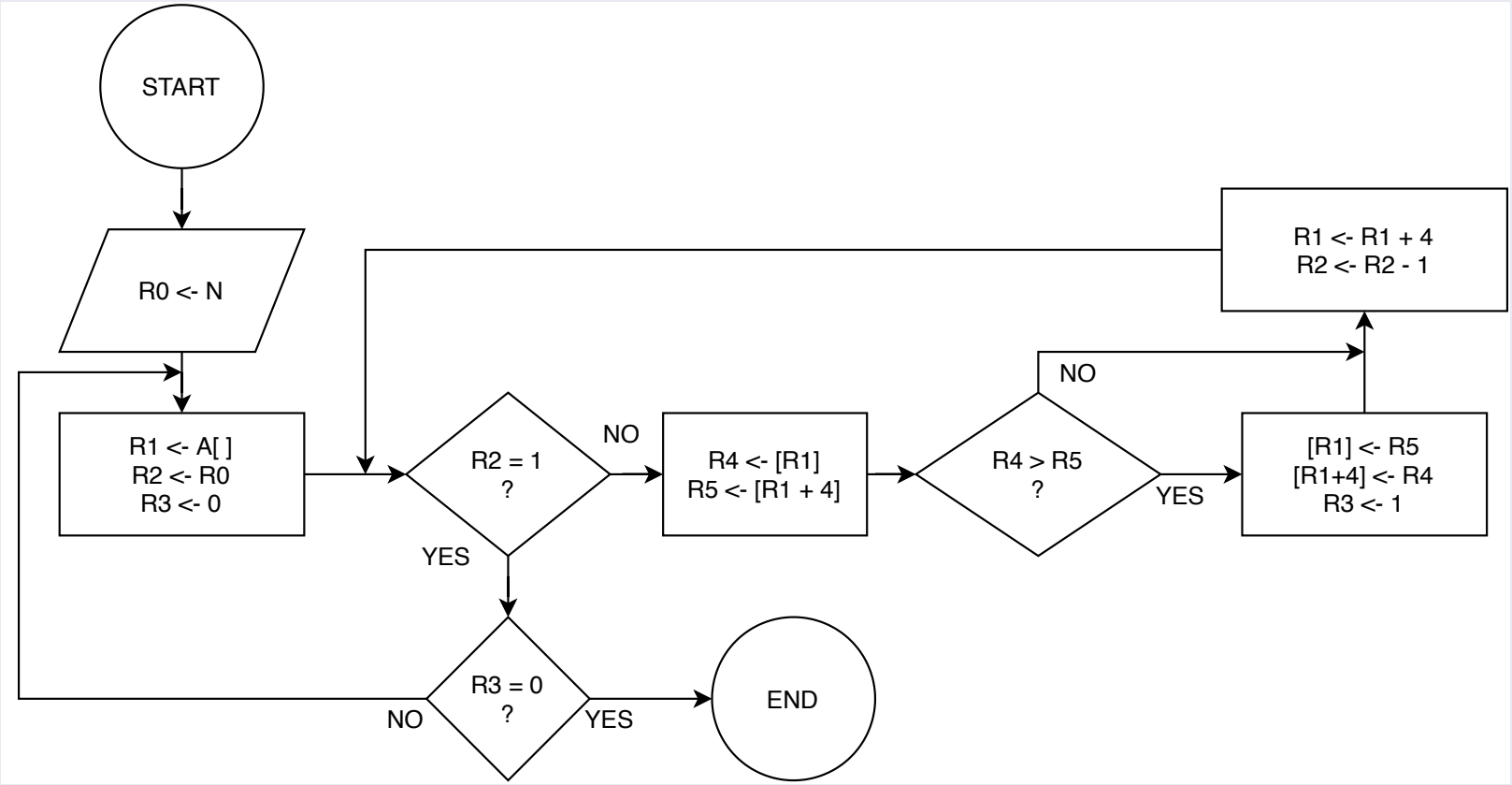
Bubble Sort

Flow Chart


START
-> N
-> A[ ]
I <- 0
COUNT <- N
FLAG <- 0
COUNT = 1
?
NO
YES
A[I] > A[I+1]
?
NO
YES
SWAP A[I], A[I+1]
FLAG <- 1
I <- I + 1
COUNT <- COUNT - 1
FLAG = 0
?
NO
YES
END

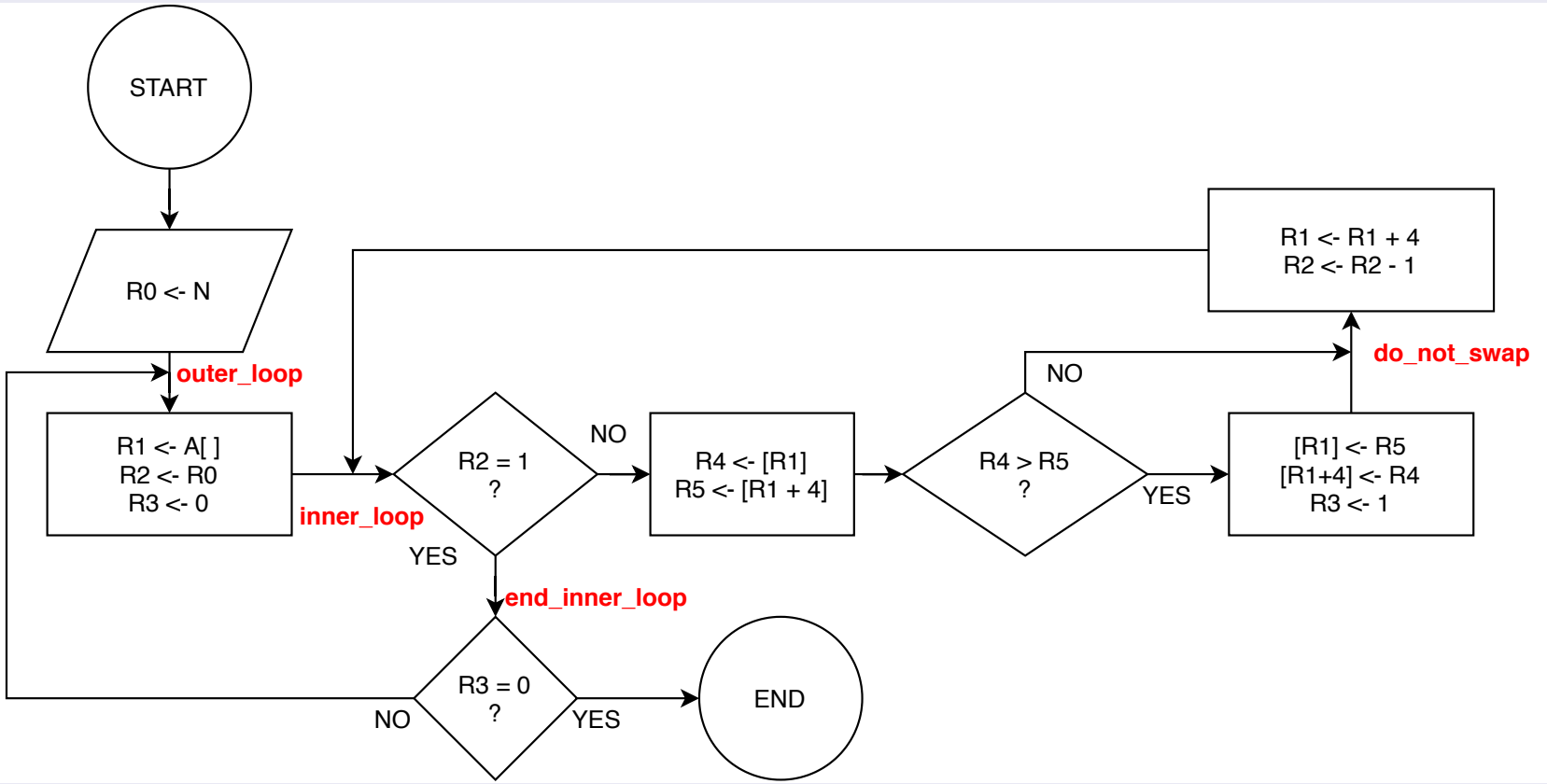
Flow Chart—Mappatura dei registri

R0←N **R1**←A[ ] **R2**←COUNT **R3**←FLAG
R4←A[I] **R5**←A[I+1]



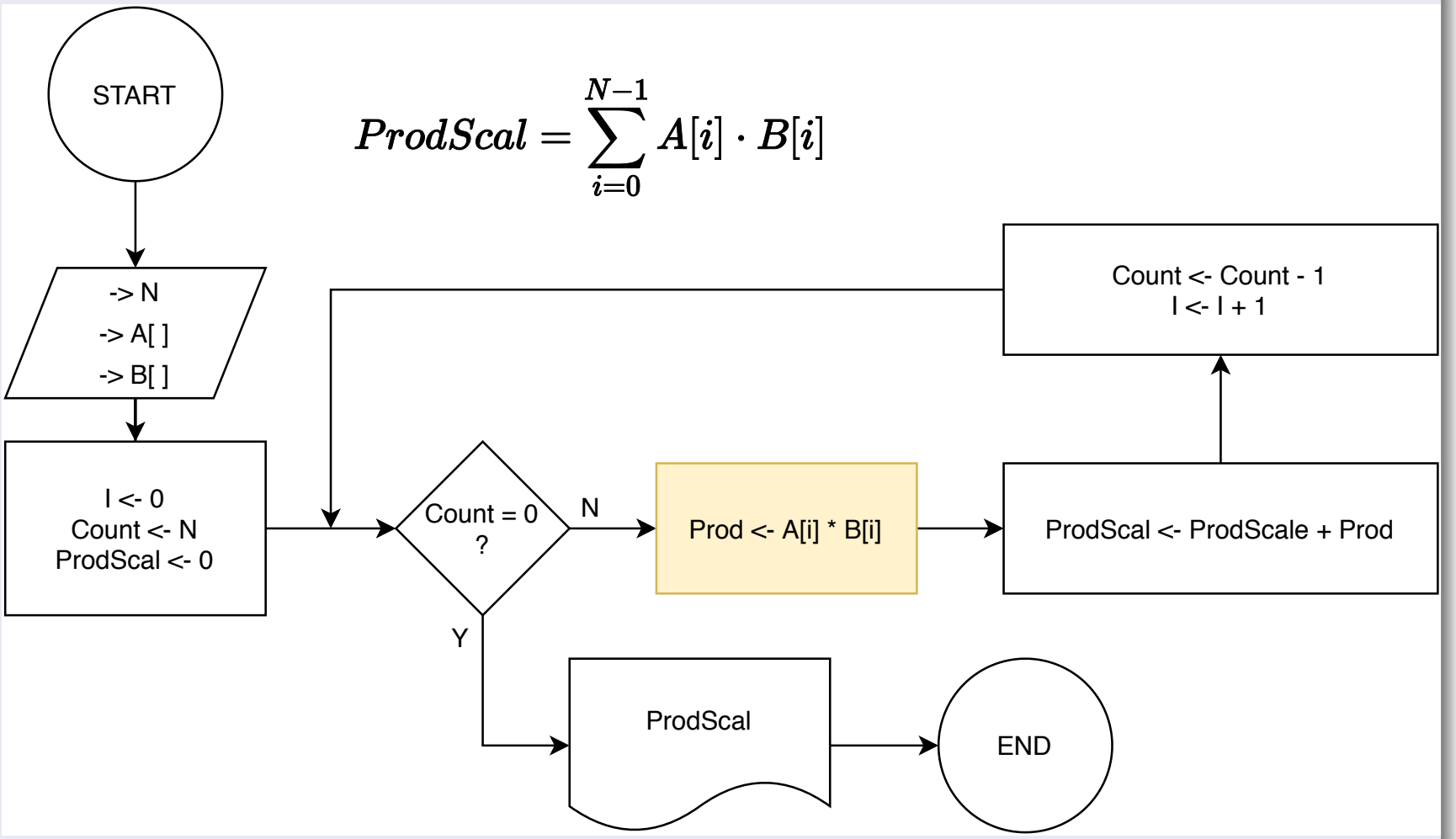
Flow Chart—Mappatura delle destinazioni (label)

R0←N **R1**←A[ ] **R2**←COUNT **R3**←FLAG
R4←A[I] **R5**←A[I+1]



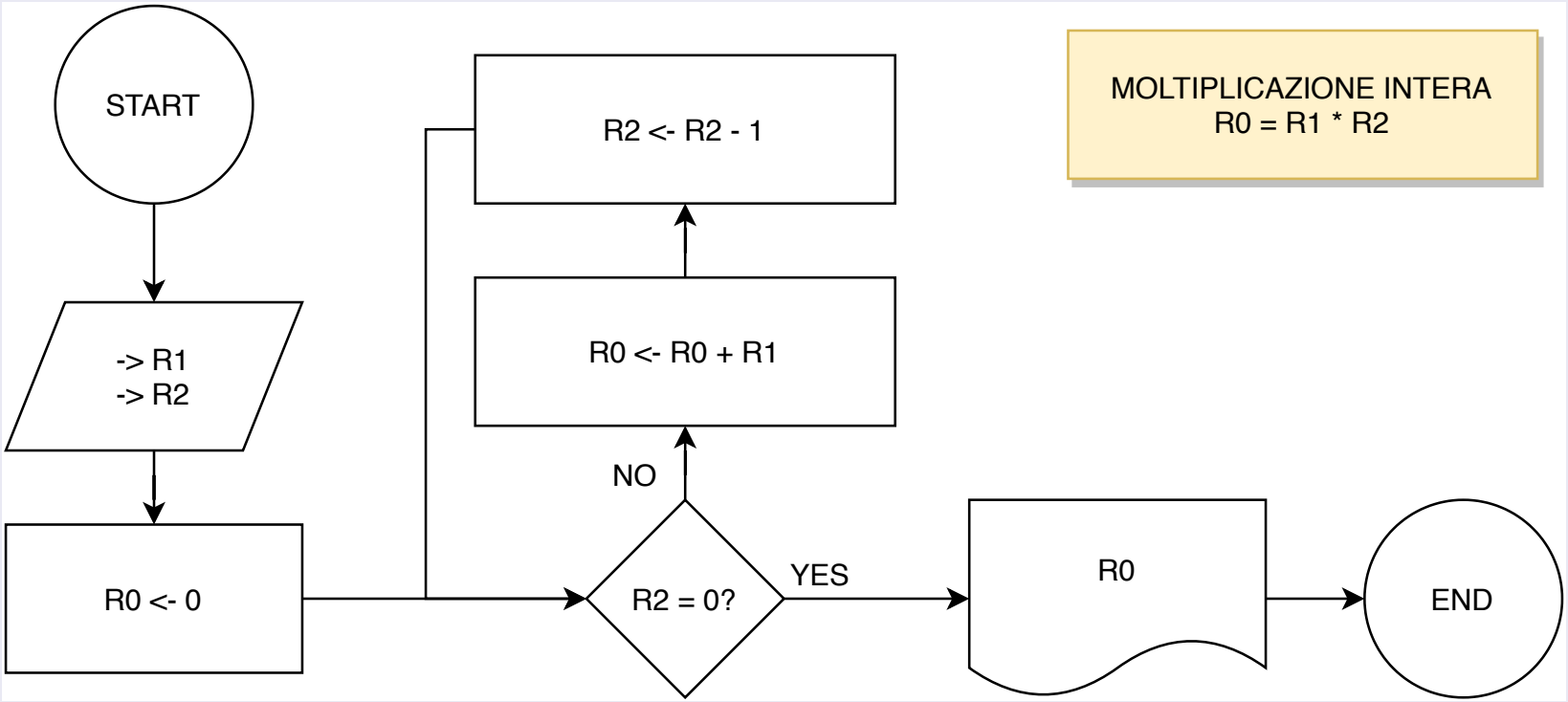
Prodotto Scalare

Flow Chart


START
$$ProdScal = \sum_{i=0}^{N-1} A[i] \cdot B[i]$$
-> N
-> A[ ]
-> B[ ]
I <- 0
Count <- N
ProdScal <- 0
Count = 0
?
N
Y
Prod <- A[i] * B[i]
ProdScal <- ProdScale + Prod
Count <- Count - 1
I <- I + 1
ProdScal
END

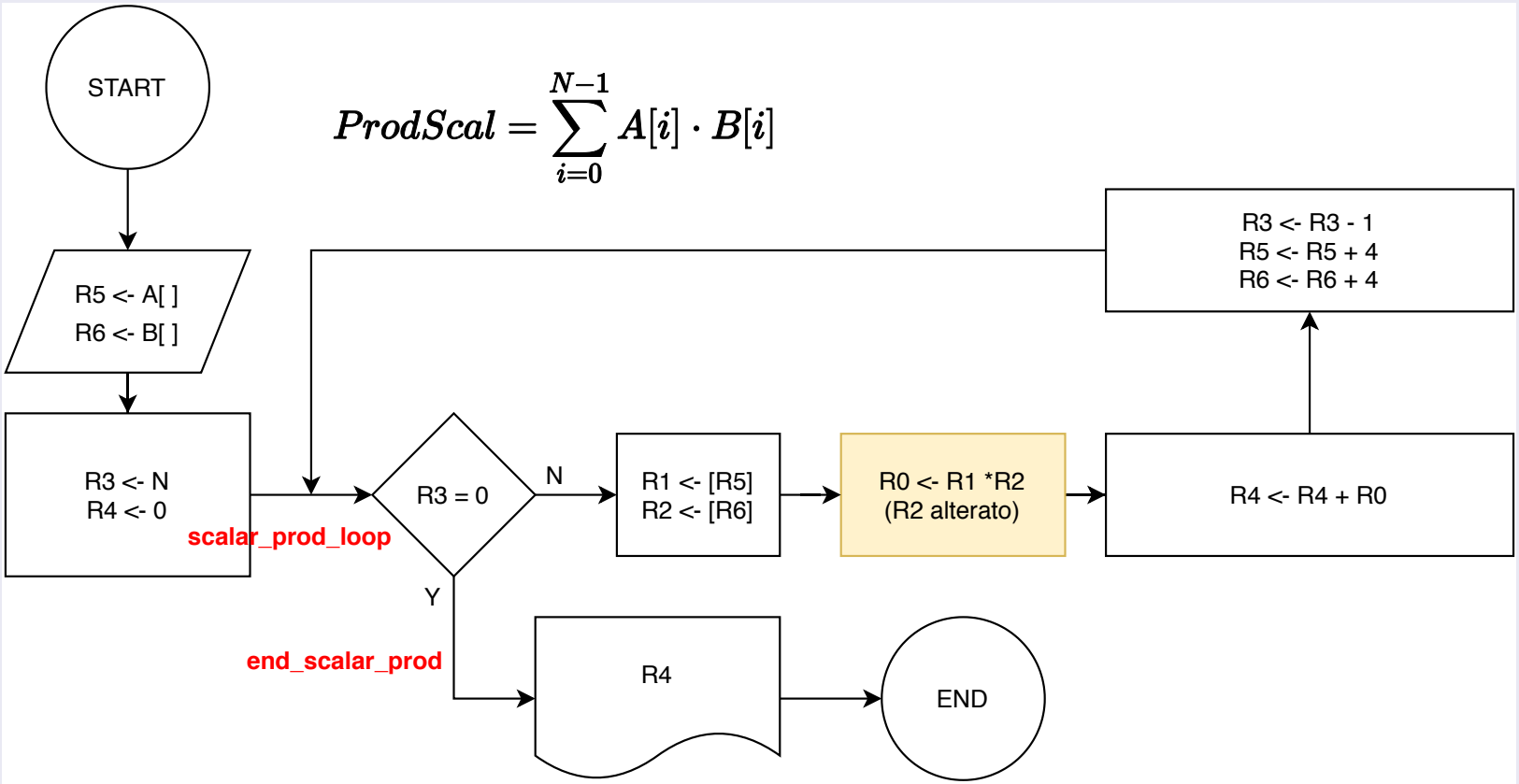
Moltiplicazione Intera

$$R0 = R1 \cdot R2 = \underbrace{R1 + R1 + \cdots + R1}_{\text{R2 volte}}$$



Flow Chart—Mappatura dei registri

R0←Prod **R1**←A[I] **R2**←B[I] **R3**←COUNT
R4←ProdScal **R5**←A[ ] **R6**←B[ ]



Ricerca di una Sequenza

Ricerca di una Sequenza

Problema

- Data una **sequenza numerica** desideriamo verifcare se essa contiene un data **sotto-sequenza** e, in caso affermativo, da quale punto parte
- **Esempio:** data la sequenza $seq = \{4, 5, 8, 10, 1, 3, 4, 25\}$, la sottosequenza $sub = \{10, 1, 3\}$ si trova in posizione 3 (a partire da 0)

Approccio

- Iteriamo sulla sequenza principale con un indice i
- Verifichiamo se la sotto-sequenza sub è presente in seq a partire da i $seq[i] == sub[0]$, $seq[i+1] == sub[1]$, $seq[i+2] == sub[2]$, ...
- In altri termini iteriamo, con indice j, su sub e verifichiamo se $seq[i+j] == sub[j]$

Ricerca di una Sequenza

Problema

```
int find_sequence(int * seq, int seq_len, int * sub, int sub_len)
{
    int i, j;
    for (i = 0; i < seq_len - sub_len;i++) {
        int found = 1;
        for (j = 0;j < sub_len;j++) {
            if (seq[i+j] != sub[j]) {
                found = 0;
                break;
            }
        }
        if (found) return i;
    }
    return -1;
}
```

Ricerca di una Sequenza

Verso l'assembly... introduzione dei "goto"

```
int find_sequence(int * seq, int seq_len,
                  int * sub, int sub_len)
{
    int i, j;
    for (i = 0; i < seq_len - sub_len;i++) {
        for (j = 0;j < sub_len;j++) {
            if (seq[i+j] != sub[j]) {
                goto skip;
            }
        }
        return i;
skip:
    }
    return -1;
}
```

Ricerca di una Sequenza

Verso l'assembly... introduzione dei puntatori

```
int find_sequence(int * seq, int seq_len,
                  int * sub, int sub_len)
{
   int i, j;
   for (i = 0; i < seq_len - sub_len;i++) {
       int * temp_seq_ptr = seq;
       int * temp_sub_ptr = sub;
       for (j = 0;j < sub_len;j++) {
           if (*temp_ptr != *temp_sub_ptr) {
               goto skip;
           }
           temp_ptr++;
           temp_sub_ptr++;
       }
       return i;
skip:
       seq++;
   }
   return -1;
}
```

Ricerca di una Sequenza

Verso l'assembly... trasformazione degli indici

```
int find_sequence(int * seq, int seq_len,
                  int * sub, int sub_len)
{
    int i, j;
    for (i = seq_len; i > sub_len;i--) {
        int * temp_seq_ptr = seq;
        int * temp_sub_ptr = sub;
        for (j = sub_len; j > 0;j--) {
            if (*temp_ptr != *temp_sub_ptr) {
                goto skip;
            }
            temp_ptr++;
            temp_sub_sub++;
        }
        return i;
skip:
        seq++;
    }
    return -1;
}
```

Ricerca di una Sequenza

Verso l'assembly... trasformazione di for in while

```
int find_sequence(int * seq, int seq_len, int * sub, int sub_len)
{
    int i, j;
    i = seq_len;
    while (i != sub_len;) {
        int * temp_seq_ptr = seq;
        int * temp_sub_ptr = sub;
        j = sub_len;
        while (j != 0) {
            if (*temp_ptr != *temp_sub_ptr)
                goto skip;
            temp_ptr++;
            temp_sub_ptr++;
            j--;
        }
        return i;
skip:
        seq++;
        i--;
    }
    return -1;
}
```

Ricerca di una Sequenza

Verso l'assembly... mappatura dei registri

```
int find_sequence(int * seq, int seq_len, int * sub, int sub_len)
{
    R0 = seq_len;
    R1 = seq;
    while (R0 != R2) {
        R6 = R1;
        R3 = sub; R2 = sub_len
        while (R2 != 0) {
            R4 = *R6; R5 = *R3;
            if (R4 != R5) goto skip;
            R6 = R6 + 4;
            R3 = R3 + 4;
            R2 = R2 - 1;
        }
        return (R1 - seq) / 4;
skip:
        R1 = R1 + 4;
        R0 = R0 - 1;
    }
    return -1;
}
```

Ricerca di una Sequenza

Implementazione in assembly—Parte 1

```
seq_a_size    dcd     10
seq_a         dcd     9,8,7,6,5,4,3,2,1,0

seq_b_size    dcd     2
seq_b         dcd     6,5

              mov     r1, #seq_a_size
              ldr     r0, [r1]
              add     r1,r1, #4

find_loop
              mov     r3,#seq_b_size
              ldr     r2,[r3]
              add     r3,r3,#4
              cmp     r0,r2
              blt     end_search_not_found
              ...
```

Ricerca di una Sequenza

Implementazione in assembly—Parte 2

```
            ...
            mov     r6, r1
find_inner_loop
            ldr     r4,[r6]
            ldr     r5,[r3]
            cmp     r4,r5
            bne     skip
            add     r6,r6,#4
            add     r3,r3,#4
            sub     r2,r2,#1
            cmp     r2,#0
            bne     find_inner_loop
            ...
```

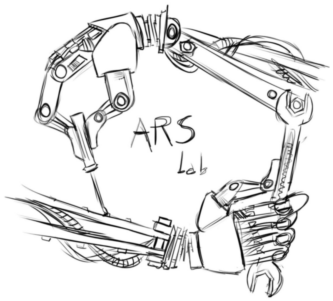
Implementazione in assembly—Parte 3

```
            ...
            sub     r0, r1, #seq_a  ;; found!!
            lsr     r0,r0, #2
            b       end_program
skip
            add     r1,r1,#4
            sub     r0,r0,#1
            b       find_loop

end_search_not_found
            mov     r0, #-1

end_program
            end
```

Bubble Sort e Prodotto Scalare in Assembly

Corrado Santoro

Dipartimento di Matematica e Informatica
santoro@dmi.unict.it



Corso di Architettura degli Elaboratori