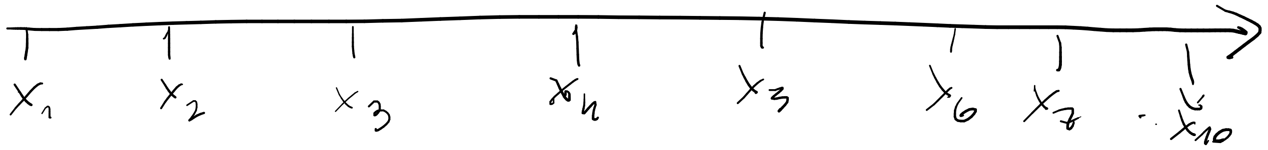
Esercizio: Sia $X = \{x_1, x_2, \dots, x_n\}$ un insieme di punti ordinati sulla retta reale.
Fornire un algoritmo greedy che determini un insieme $I$ di cardinalità minima di intervalli chiusi di ampiezza unitaria ($[a,b] \in I \Rightarrow b-a=1$) tale che $\forall x_i \in X \; \exists j \in I$ tale che $x_i \in j$.

$[1/2]$
$[1/2]$ RIESCO
$[1/2]$ UNO
$[1/2]$ CHE PIÙ!

$\cancel{[1 \dots 8]}$

INSIEME → CARD. MINIMA

(1)




MIN-COVER (X) (GREEDY)
N = LENGTH (X)
OPT = $\{x_1\}$
FOR i = N DOWNTO 2 $(a, b \in I \mid |b-a| = 1)$
IF $|N_i - OPT| == 1$
OPT = OPT $\cup \{N_i\}$

RETURN OPT → PARTIRE DALLA FINE (OPPURE FARE PARTIRE DALL'INIZIO)

√ 1) da sx a dx, inizia un nuovo intervallo nel primo punto non coperto

```
MIN_COVER(X)
  n = length(X)
  C = {[x1, x1+1]}
  last = 1
  for i = 2 to n do
    if xi > xlast + 1 then
      C = C ∪ {[xi, xi+1]}
      last = i
  return C
```

FUORI ORDINE... (INDICI NON ORDINATI)

---

② PROPRIETÀ DI SOTTOSTRUTTURA OTTIMA

(s1) $\sigma' = \{x_j, x_{j+1}\}$

$\underset{OPT}{\underline{C}} = C \cup \{\underline{\sigma^*}\} = C$

$C' = C \setminus \sigma^* = C' \neq C$

GREEDY ALGO = SOL. OPT = $\{x_i, x_i + 1\} \dots m$

**Esercizio 2** (9 punti) Per $n > 0$, siano dati due vettori a componenti intere $\mathbf{a}, \mathbf{b} \in \mathbf{Z}^n$. Si consideri la quantità $c(i,j)$, con $0 \le i \le j \le n-1$, definita come segue:

$$c(i,j) = \begin{cases} a_i & \text{se } 0 < i \le n-1 \text{ e } j = n-1, \\ b_j & \text{se } i = 0 \text{ e } 0 \le j \le n-1, \\ c(i-1, j-1) + c(i, j+1) & 0 < i \le j < n-1. \end{cases}$$

Si vuole calcolare la quantità $m = \min\{c(i,j) : 0 \le i \le j \le n-1\}$.

1. Si fornisca il codice di un algoritmo iterativo bottom-up per il calcolo di $m$.
2. Si valuti la complessità esatta dell'algoritmo, associando costo unitario alle somme tra numeri interi (escluse quelle tra indici) e costo nullo a tutte le altre operazioni.

1 SOLA FUNZIONE

PER COMPLESSITÀ MINORE, NE FAI UNO SOLO

```
COMPUTE (A, B)
N = LENGTH (A)
FOR i=1 TO N-1
    C[i, N-1] = A[i]
    C[0, i] = B[i]
```

$\Rightarrow 0 \le j \le n-1 \rightarrow$ HA SENSO ALTRO CICLO SOLO PER L'INDICE ...

```
FOR i = 1 TO N-1                 FOR J = 0 TO N-1
    C[i, N-1] = A[i]                 C[0, J] = B[J]
    M = MIN (C[i, N-1], M)           M = MIN (C[0, J], M)
```

C(i-1, J-1) + C(i, J+1) $\leftarrow$ $0 < i \le j < n-1$

```
FOR i = 1 TO N-2
    FOR J = N-2 DOWNTO i
        C[i, J] = C[i-1, J-1] + C[i, J+1]
        M = MIN (M, C[i, J])

RETURN M
```

1. Date le dipendenze tra gli indici nella ricorrenza, un modo corretto di riempire la tabella è attraverso una scansione in cui calcoliamo gli elementi in ordine crescente di indice di riga e, per ogni riga, in ordine decrescente di indice di colonna. Il codice è il seguente.

```
COMPUTE(a,b)
n <- length(a)
m = +infinito
for i=1 to n-1 do
   C[i,n-1] <- a_i
   m <- MIN(m,C[i,n-1])
for j=0 to n-1 do
   C[0,j] <- b_j
   m <- MIN(m,C[0,j])
for i=1 to n-2 do
   for j=n-2 downto i do
      C[i,j] <- C[i-1,j-1] + C[i,j+1]
      m <- MIN(m,C[i,j])
return m
```

1

$\sum\sum \ldots \Rightarrow \sum_{i=1}^{n-2} \sum_{j=i}^{n-2} 1 =$

COMPLESSITÀ ESATTA!

$\sum_{i=1}^{n-2} (\underbrace{n-i-1}_{K}) =$

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