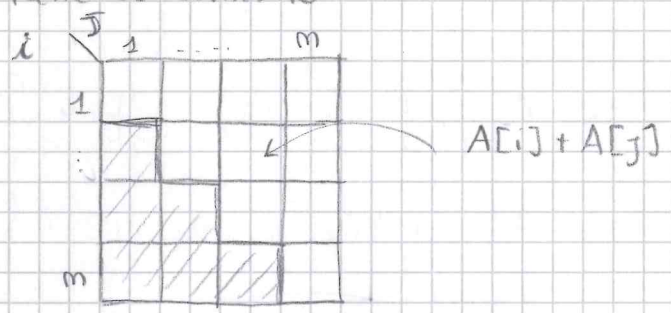
ESERCIZIO: Realizzare una funzione Sum(A, key) che dato un array A ed un intero key verifica se key è esprimibile come somma di elementi di A

<u>Soluzione banale</u>:

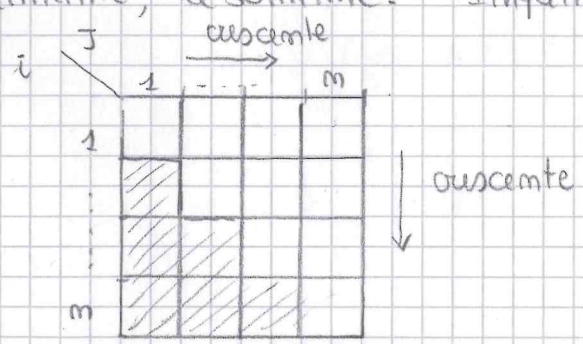
considero tutte le somme




```
Sum (A, key)
    i = 1
    j = 1
    while (i ≤ m) and (A[i] + A[j] ≠ key)
        if j = m
            i = i + 1
            j = i
        else
            j = j + 1
    return (i ≤ m)
```

costo $\Theta(m^2)$

* Si può fare di meglio? Sì, ordinando l'array e quindi, implicitamente, le somme. Infatti, se A è ordinato

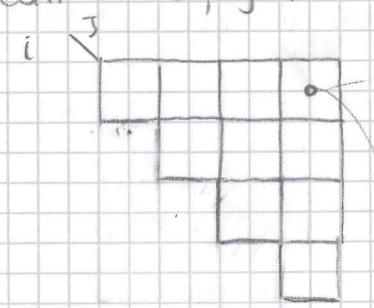


anche le somme A[i] + A[j] spostandosi a dx e in basso crescono

Σ

posso sfruttare utilmente questo fatto

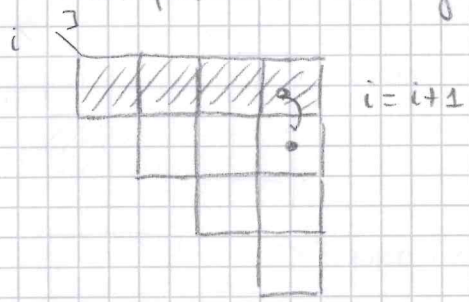
parto con $i = 1$, $j = n$



se $A[i] + A[j] = key$ fine!

se $A[i] + A[j] < key$

vale per l'intera riga



se $> key$ ⇝ elimino la colonna

$j = j - 1$

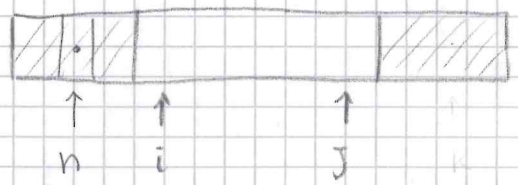
in questo modo restringo lo spazio di ricerca concludendo in tempo lineare

```
Sum (A, key)
   Sort (A)                (in place)
   i = 1
   j = A.length

   while (i ≤ j) and (A[i] + A[j] ≠ key)
       if A[i] + A[j] < key
           i = i + 1
       else
           j = j - 1

   return i ≤ j
```

Qual è l'invariante?



(a) $\forall h \in [1\ i-1] \quad \forall k \in [h, m] \qquad A[h] + A[k] \neq key$

(ogni elemento della parte scartata a sx sommato a qualunque elemento successivo non produce key)

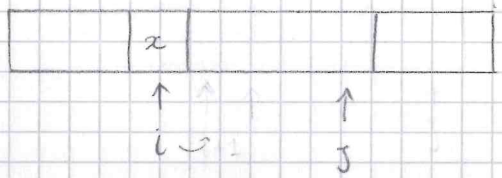
(b) $\forall k \in [j+1, m] \quad \forall h \in [1, k] \qquad A[h] + A[k] \neq key$

(duale)

→ inizializzazione: banalmente vero all'inizio

→ mantenimento:

- se $A[i] + A[j] < key$ ⇝ quindi sposto $i \rightsquigarrow i+1$



(a) resta vera, infatti

per $k \in [i, j] \qquad A[k] \leq A[j]$

$\Rightarrow A[i] + A[k] \leq A[i] + A[j] < key$

per $k \in [j+1, m] \qquad i \in [1..k-1]$ quindi per (b)

$A[i] + A[k] \neq key$

(b) non influenzata

- se $A[i] + A[j] > key$

duale

⇝

Conclusione :

- se $A[i]+A[j] = key$ ok!

- se $i > j$ $\rightsquigarrow$ $i = j+1$

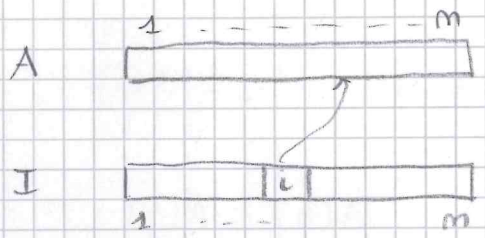
  (a) e (b) implicano che la somma non si può ottenere precisamente presi $h, k \in [1..m]$ $h \leq k$

  - $h \leq i$ per (a) $A[h] + A[k] \neq key$
  - altrimenti $h > i \Rightarrow k \geq j \Rightarrow$ per (b) $A[h]+A[k] \neq key$

* Complessità : $O(m)$ (includendo l'ordinamento $O(m \log m)$)

→ se volessimo gli indici i, j dell'array originale?

idea : costruisci un array indice $I[1..m]$ e lavora su quello!



tale che $i \leq j \Rightarrow A[I[i]] \leq A[I[j]]$

Array delle occorrenze

Se ho un limite inferiore per i valori in A (ad es. sono tutti ≥ 0) posso mantenere un array delle occorrenze "utili" che ricordi la presenza in A di elementi che possono concorrere a formare una somma = key (quindi nello specifico $\leq$ key)

con $V[v] = \text{true}$ sse esiste $i \in [1,m]$ t.c. $A[i] = v$

la funzione può quindi essere la seguente:

```
sum (A , key)
    V[0..key] ← false

    i = 1
    found = false

    while (i ≤ m) and not found
        if (A[i] ≤ key)
            V[ A[i] ] = true
            found = V[ key - A[i] ]

        i++

    return found
```