

## Partizione di un insieme di interi

$A = \{a_1, a_2, \dots, a_n\}$  insieme di interi positivi  
di somma pari

$$\text{Somma}(A) = \sum_{i=1}^n a_i = 2s$$

Determinare, se esiste, un sottoinsieme  $A' \subseteq A$   
t.c.  $\text{Somma}(A') = s$ .

### ALGORITMO ENUMERATIVO

si generano tutti i sottoinsiemi di  $A$ .

Costo in tempo  $O(2^n \times n)$

### ALGORITMO di PROGRAMMAZIONE DINAMICA

① Sottoproblemi

$\Pi_{ij}$ :  $\exists A' \subseteq \{a_1, \dots, a_i\}$  t.c.  $\text{Somma}(A') = j$ ?

matrice di PD:  $(n+1) \times (s+1)$

~~partizioni~~ ~~partizioni~~

$\text{part}[i, j] = \text{true} \Leftrightarrow \exists A' \subseteq \{a_1, \dots, a_i\}$ t.c. $\text{Somma}(A') = j$ $0 \leq i \leq n$ $0 \leq j \leq s$
---

② Sottoproblemi elementari

$$\text{parti}(0,0) = \text{true}$$

③ Regola Ricorsiva

$$\text{parti}(i,j) = \begin{cases} \text{true} & i=0, j=0 \\ \text{true} & i>0, \text{parti}(i-1,j)=\text{true} \\ \text{true} & i>0, j \geq a_i \\ & \text{parti}(i-1, j-a_i)=\text{true} \\ \text{false} & \text{altrimenti} \end{cases}$$

per  $i > 0$

$$\text{parti}(i,j) = \text{true} \Leftrightarrow$$

$$\text{parti}(i-1, j) = \text{true}$$

OPPURE

$$j \geq a_i \text{ AND } \text{parti}(i-1, j-a_i) = \text{true}$$

④ ~~reflexiva~~ ~~partita~~

$$\text{parti}(n, s) = \text{true} \Leftrightarrow$$

$$\exists A' \subseteq A = \{a_1, \dots, a_n\} \\ \text{di somma } s$$

Partizione (a) // a: array di n interi positivi

somma = 0

for i = 1 to n  
    somma = somma + a[i]

if (somma % 2 == 1) return false;

s = somma / 2

parti = nuova matrice (n+1) x (s+1)

for i = 0 to n  
    parti[i, 0] = true

for i = 0 to n  
    for j = 1 to s  
        parti[i, j] = false

for i = 1 to n {  
    for j = 1 to s {  
        if (parti[i-1, j]) parti[i, j] = true;  
        else if (j ≥ a[i] && parti[i-1, j-a[i]])  
            parti[i, j] = true;  
    }

}  
return parti[n, s]

$$T(n, s) = \Theta(n \times s)$$

Algoritmo pseudo polinomiale

# Ricostruzione del sottoinsieme di somma S

Sottoinsieme (a, parti) // a: array  
parti: matrice di PD

```
if (parti[n, s]) {  
    i = n;  
    j = s;  
    while (i > 0) {  
        if (!parti[i-1, j]) {  
            print a[i];  
            j = j - a[i];  
        }  
        i--;  
    }  
}
```

else print "Non sottoinsieme di somma S"

$$T(n) = O(n)$$

ESEMPIO

A = {2, 4, 1, 3}

S = 5

Q:

	0	1	2	3	4	5
0	t	f	f	f	f	f
2	t	f	t	f	f	f
4	t	f	t	f	t	f
1	t	t	t	t	t	t
3	t	t	t	t	t	t

PRINT: 1, 4