

# Partial Sum

Vogliamo implementare una struttura dati che supporti le seguenti operazioni:

**Init** che prende una lista  $A$  e restituisce una rappresentazione  $\text{Repr}$  di  $A$ .

**Lookup** che prende in input  $i < j$  e  $\text{Repr}$  restituisce la somma  $A[i] + A[i + 1] + \dots + A[j - 1]$ .

**Set** che prende in input  $k$  e  $\text{val}$  ed aggiorna la rappresentazione  $\text{Repr}$ .

# Partial Sum

Descriviamo un'implementazione che, per una lista di  $N$  elementi, usa spazio  $O(N)$  e le operazioni prendono tempo

Init	$O(N)$
Lookup	$O(\log N)$
Set	$O(\log N)$

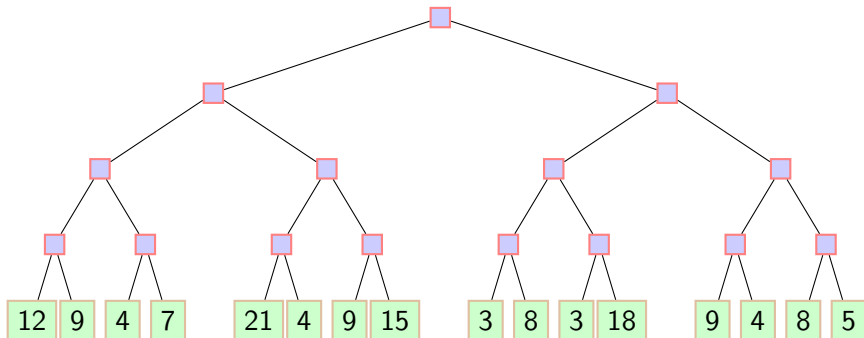
# Init: costruire la rappresentazione

Precomputiamo le risposte alle seguenti query:

- ▶  $(0, 1), (1, 2), \dots, (N - 1, N)$ :  
*tutte le query  $(i, i + 1)$  di lunghezza 1, per  $i$  multiplo di 1.*
- ▶  $(0, 2), (2, 4), \dots, (N - 2, N)$ :  
*tutte le query  $(i, i + 2)$  di lunghezza 2, per  $i$  multiplo di 2.*
- ▶  $(0, 4), (4, 8), \dots, (N - 4, N)$ :  
*tutte le query  $(i, i + 4)$  di lunghezza 4, per  $i$  multiplo di 4.*
- ▶ .....
- ▶  $(0, N)$ :  
*tutte le query  $(i, i + N)$  di lunghezza  $N$ , per  $i$  multiplo di  $N$ .*

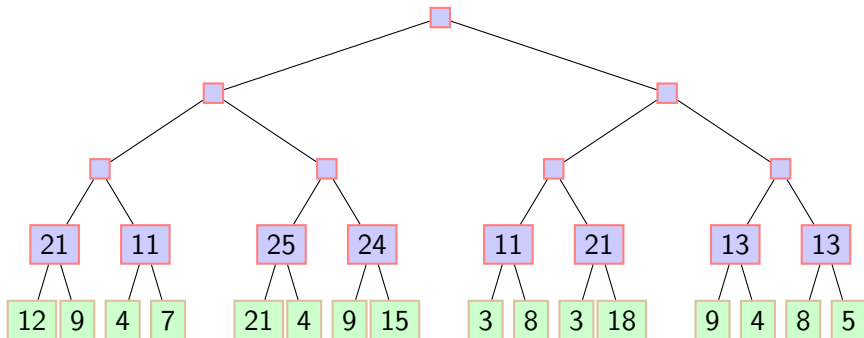
[12, 9, 4, 7, 21, 4, 9, 15, 3, 8, 3, 18, 9, 4, 8, 5]

Risposte a  $\text{Lookup}(i, i + 1)$ , per  $i$  multiplo di 1



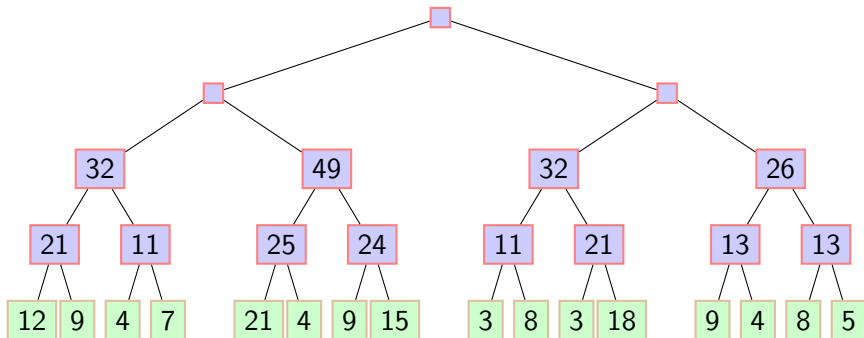
[12, 9, 4, 7, 21, 4, 9, 15, 3, 8, 3, 18, 9, 4, 8, 5]

Risposte a  $\text{Lookup}(i, i + 2)$ , per  $i$  multiplo di 2



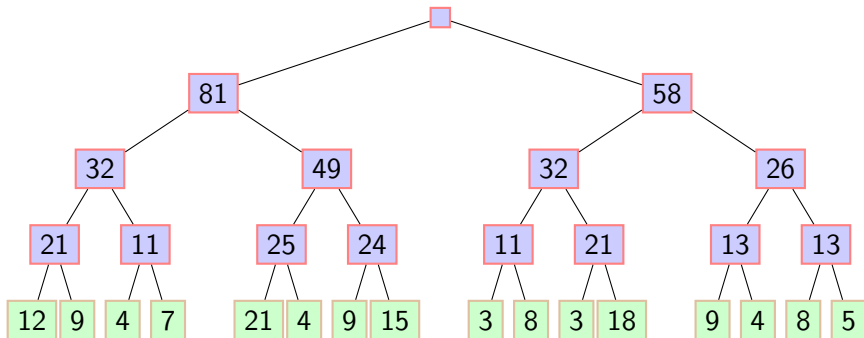
[12, 9, 4, 7, 21, 4, 9, 15, 3, 8, 3, 18, 9, 4, 8, 5]

Risposte a  $\text{Lookup}(i, i + 4)$ , per  $i$  multiplo di 4



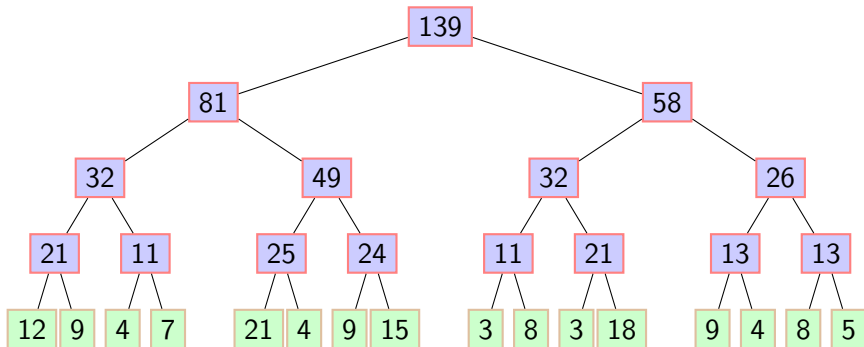
[12, 9, 4, 7, 21, 4, 9, 15, 3, 8, 3, 18, 9, 4, 8, 5]

Risposte a  $\text{Lookup}(i, i + 8)$ , per  $i$  multiplo di 8



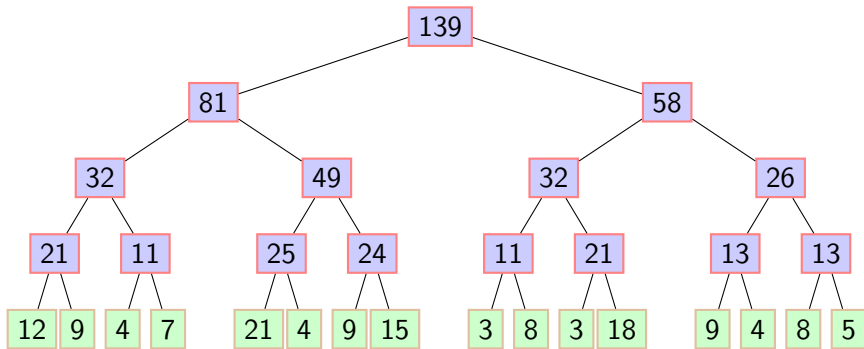
[12, 9, 4, 7, 21, 4, 9, 15, 3, 8, 3, 18, 9, 4, 8, 5]

Risposte a  $\text{Lookup}(i, i + 16)$ , per  $i$  multiplo di 16

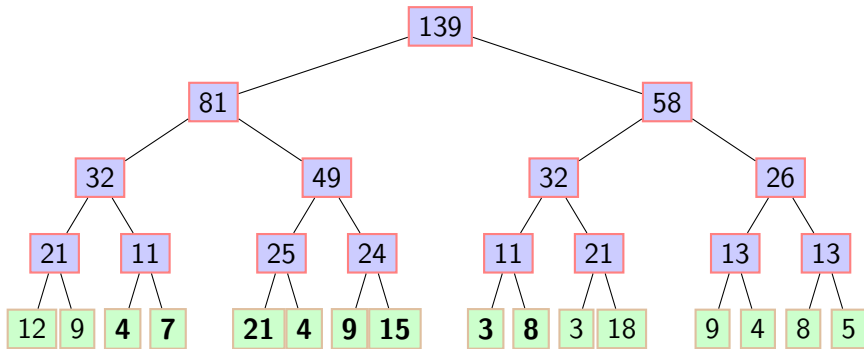




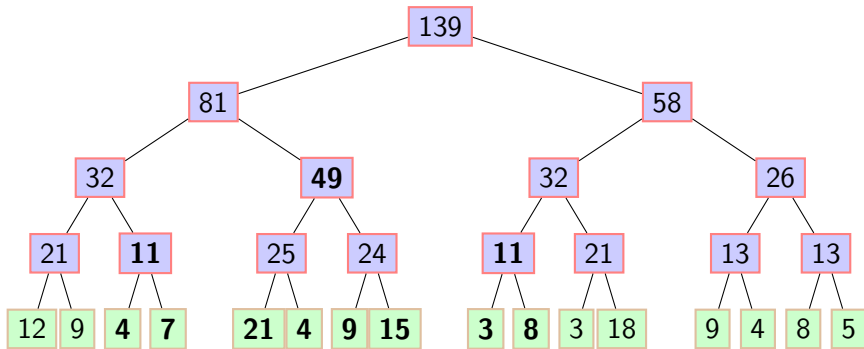
$$\text{Lookup}(2, 10) = \text{Lookup}(2, 4) + \text{Lookup}(4, 8) + \text{Lookup}(8, 10)$$



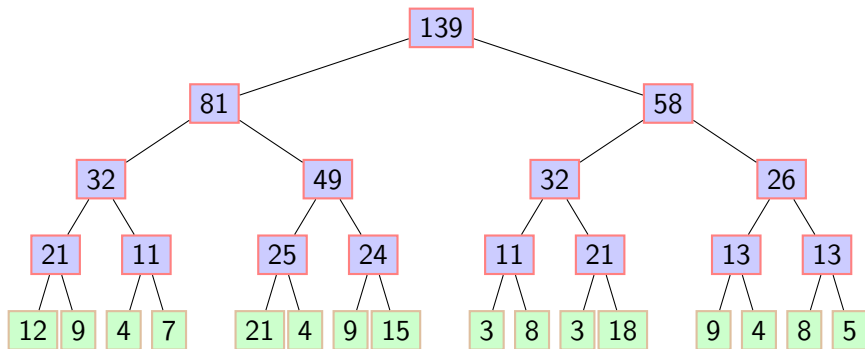
$$\text{Lookup}(2, 10) = \text{Lookup}(2, 4) + \text{Lookup}(4, 8) + \text{Lookup}(8, 10)$$



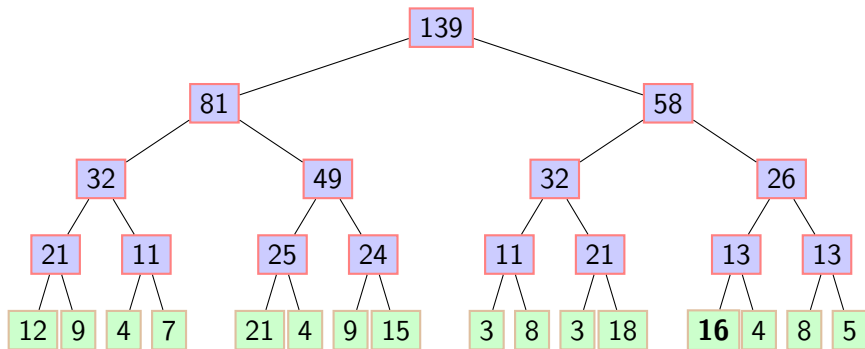
$$\text{Lookup}(2, 10) = \text{Lookup}(2, 4) + \text{Lookup}(4, 8) + \text{Lookup}(8, 10)$$



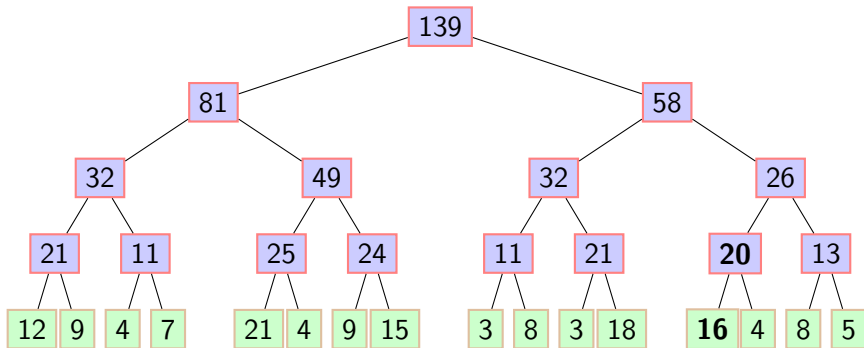
Set(12, 16)



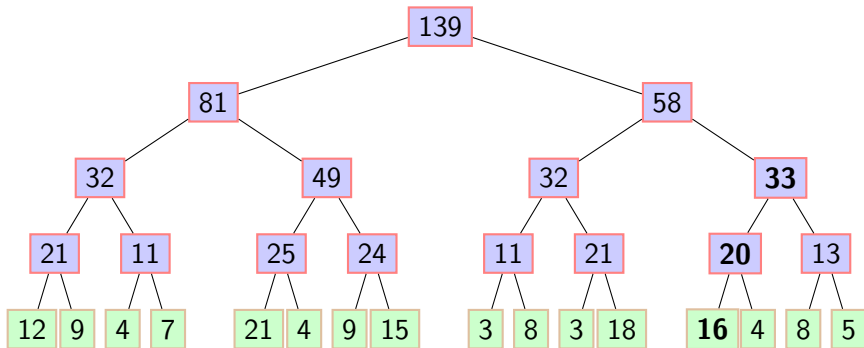
Set(12, 16)



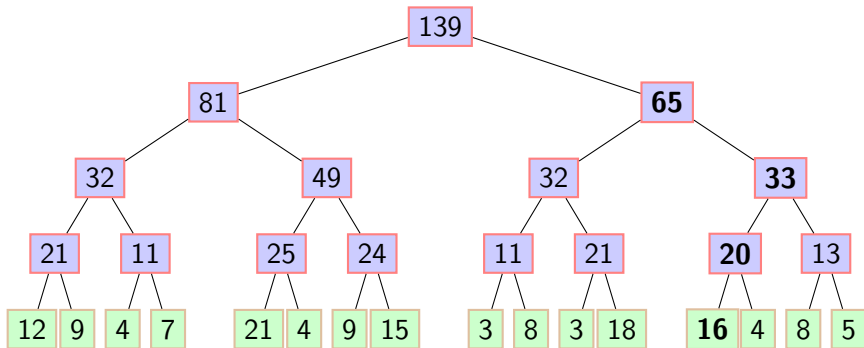
Set(12, 16)



Set(12, 16)

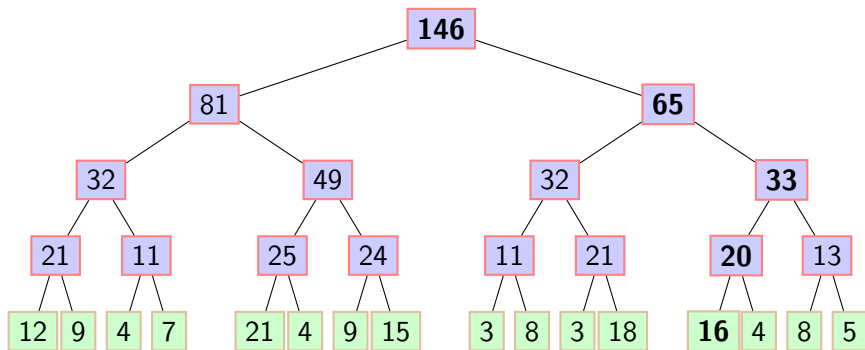


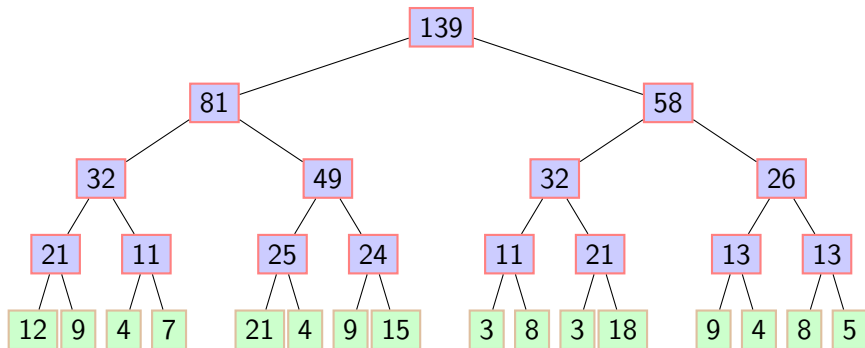
Set(12, 16)

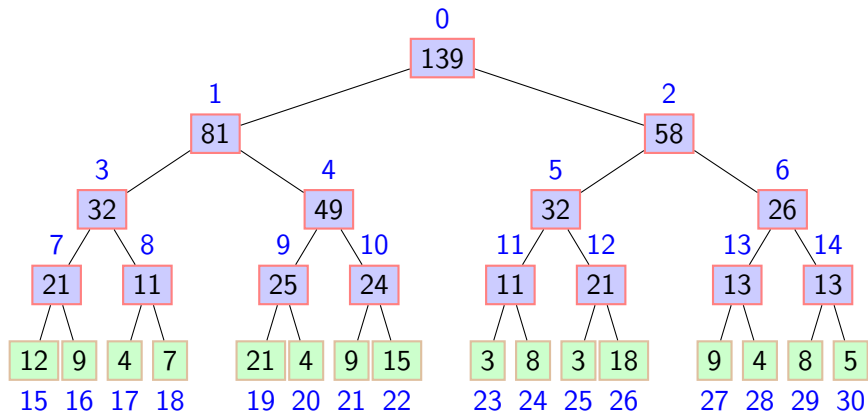




Set(12, 16)







# Costruire Repr

## Primo tentativo: Insert in lista

```
def __init__(self,B):
    self.N=len(B)
    self.A=B.copy()
    start=len(B)-1
    while start>0:
        end=0
        while start>end:
            t=self.A[start]+self.A[start-1]
            start=start-2 #consumo due elementi
            self.A.insert(0,t)
            start=start+1
            end=end+1 #annullo l'effetto della insert su start e end
```

# Construire Repr

Secondo tentativo: Append in lista

```
def __init__(self,B):
    self.A=B.copy()
    self.A.reverse()
    self.N=len(B)
    start=0
    end=self.N
    while start<end-1:
        for i in range(start,end,2):
            self.A.append(self.A[i]+self.A[i+1])
            l=(end-start)//2
            start=end
            end=end+l
    self.A.reverse()
```