

**Corso di Laurea Magistrale in Ingegneria Informatica**

**CORSO DI  
ALGORITMI E STRUTTURE DATI**

**Prof. ROBERTO PIETRANTUONO**

**Prova in itinere**

**Indicazioni**

Si consegna un file in **formato editabile (.txt, .docx, .rtf, etc.)** nominandolo "*CognomeNome*", in cui è riportata l'implementazione (nel linguaggio scelto) seguita da una indicazione della complessità temporale dell'algoritmo implementato (complessità nel caso peggiore, è sufficiente il limite superiore  $O(f(n))$ ). Se si utilizzano librerie di cui non si conosce la complessità, lo si indichi nella spiegazione (ad esempio, "la complessità è  $O(n \log n)$  al netto della complessità dell'algoritmo  $x$ , che è non nota"). Se si utilizza la randomizzazione, si indichi anche il tempo di esecuzione atteso.

**PROBLEMA 1**

Si implementi un algoritmo seguendo l'approccio *Divide et Impera* per contare le occorrenze di un intero  $K$  in un array non ordinato  $A[]$ .

**INPUT**

L'input inizia con un numero intero che indica il numero di casi di test. Ogni riga successiva è un caso di test. Il primo numero della riga indica  $K$ . Il secondo numero indica la dimensione dell'array. I numeri dal terzo in poi costituiscono gli elementi dell'array  $A$ .

**OUTPUT**

Si stampi una riga per ogni caso di test, riportante il conteggio delle occorrenze

**Sample Input**

```
2
1 7 1 1 2 2 2 2 3
4 7 1 1 2 2 2 2 3
```

**Sample Output**

```
2
0
```

**PROBLEMA 2**

Sia data una tripla di interi  $S$ ,  $N$  e  $P$ .  $S$  indica una somma.  $P$  indica un numero primo. Si trovino tutti gli  $N$  numeri primi strettamente maggiori di  $P$  tali che la loro somma è uguale ad  $S$ . Possono esistere più sequenze che soddisfano tale condizione.

*Suggerimenti:*

- Il problema si può risolvere utilizzando il *Backtracking*.
- Una soluzione è una o più sequenze di numeri che soddisfano il criterio (somma =  $S$ , lunghezza sequenza =  $N$ , numeri della sequenza maggiori di  $P$ )
- Per costruire una soluzione, le alternative tra cui scegliere per aggiungere un nuovo numero alla sequenza sono i numeri primi maggiori di  $P$  e minori o uguali di  $S$ .
- Se si considerano solo i numeri maggiori di  $P$  e minori o uguali di  $S$ , allora per accettare una sequenza come soluzione, i vincoli da controllare sono due: la somma deve essere  $S$ , e la lunghezza deve essere  $N$

- *Può essere comodo avvalersi di funzioni ausiliarie (per esempio per ottenere la sequenza di numeri primi tra  $P$  ed  $S$ )*

## **INPUT**

L'input inizia con un numero intero che indica il numero di casi di test. Ogni riga successiva è un caso di test. Il primo numero della riga indica  $S$ , il secondo numero indica  $N$ , il terzo indica  $P$ .

## **OUTPUT**

Per ogni caso di test si stampi dapprima una riga "CASO DI TEST  $n$ " dove " $n$ " indica il caso di test che si sta considerando. Seguita dalle righe che rappresentano le sequenze cercate.

### **Sample Input**

```
63 3 5
23 3 2
17 1 5
```

### **Sample Output**

```
CASO DI TEST 1
7 13 43
7 19 37
11 23 29
13 19 31
CASO DI TEST 2
3 7 13
5 7 11
CASO DI TEST 3
17
```