

Problema 1

[Punti: (a) 4/30; (b) 3/30]

Si considerino i metodi Java di seguito illustrati, in cui si fa uso della ricorsione multipla.

```

static void comb(int[] a, int i, int max) {
    if(i < 0) {
        for(int h = 0; h < a.length; h++)
            System.out.print((char)('a'+a[h]));
        System.out.print("\n");
        return;
    }
    for(int v = max; v >= i; v--) {
        a[i] = v;
        comb(a, i-1, v-1);
    }
}

static void comb(int[] a, int n) { // a.length <= n
    comb(a, a.length-1, n - 1);
    return;
}

```

Sviluppare, *argomentando adeguatamente* (il 50% del punteggio dell'esercizio sarà sulle argomentazioni addotte), quanto segue:

- Determinare il migliore upper bound possibile dell'algoritmo `comb(int[], int)`, in funzione della dimensione dell'input.
- Simulare l'esecuzione dell'algoritmo con chiamata `comb(a, 5)`, essendo `a` un array di 2 `int`. In particolare, determinare l'output dell'algoritmo.

Problema 2

[Punti: 9/30]

Dato un albero binario t a chiavi intere positive: denotiamo con $\tau(v)$ il sottoalbero con radice v , con $\text{left}(v)$ e $\text{right}(v)$, rispettivamente, il figlio sinistro e destro di v e con $k(v)$ la chiave contenuta in v ; indichiamo con $w(v)$ la somma delle chiavi nel sottoalbero di radice v , ovvero $w(v) = \sum_{u \in \tau(v)} k(u)$ (essendo $w(\text{null}) = 0$); definiamo sui vertici dell'albero la funzione $\pi(v) = |w(\text{left}(v)) - w(\text{right}(v))|$.

L'albero t è detto *albero-bello* se e solo se per tutti i suoi vertici v vale $k(v) \geq \pi(v)$. In altre parole, in un albero-bello ciascun vertice contiene una chiave non inferiore al valore assoluto della differenza fra la somma di tutte le chiavi nel sottoalbero destro e la somma di tutte le chiavi nel sottoalbero sinistro.

Ciò premesso, realizzare un metodo Java, o funzione C, `testaBello` che con input un albero binario t con chiavi intere positive restituisce 1 se t è un albero-bello, 0 altrimenti. L'algoritmo deve avere costo computazionale $\Theta(n)$. Che tipo di visita è stata eseguita?

Problema 3

[Punti: 8/30]

Definire un algoritmo efficiente per risolvere il seguente problema: dato un insieme S di n interi differenti e dato un intero k , determinare il k -esimo elemento di S in ordine crescente (senza ordinare S). Determinare il costo computazionale dell'algoritmo. (Ispirarsi al passo di partizionamento del QuickSort).

Problema 4

[Punti: 9/30]

L'arcipelago di Isolandia è formato da migliaia di isole, nelle quali sono collocati N porti (circa 400). Due differenti compagnie di battelli, la Lumaconi e la Saettanti, mettono a disposizione vari collegamenti (bidirezionali) fra coppie di porti (non c'è garanzia che tutti i porti siano raggiungibili da ogni porto). La compagnia Lumaconi utilizza battelli lenti con biglietti economici, mentre la compagnia Saettanti utilizza battelli velocissimi con biglietti costosi.

Il sig. Poverelli, che abita presso il porto P , ha urgenza di recarsi al porto Q ed, essendo il suo budget limitato, vorrebbe usufruire solo dei servizi della Lumaconi, ma non è sicuro che possa recarsi da P a Q , anche facendo più scali, utilizzando solo battelli della Lumaconi. Data l'urgenza del suo viaggio, utilizzerà, se proprio necessario, anche battelli della Saettanti.

Assumendo noti tutti i collegamenti porto-porto offerti dalle due compagnie e i relativi costi dei biglietti, risolvere i seguenti punti:

- Formulare il problema del sig. Poverelli (raggiungere Q da P spendendo il minimo possibile) come problema su grafi. A tal fine occorre dapprima definire il/i grafo/i a cui si farà riferimento, chiarendo come sia/siano collegato/i al problema in esame; successivamente va definito il problema del sig. Poverelli come problema sul/i grafo/i definito/i, utilizzando una terminologia appropriata.
- Definire un algoritmo (pseudo-codice) per la soluzione generale del problema: determinare una sequenza di battelli che consenta di raggiungere il porto Q partendo da P *minimizzando il costo*.
- Con riferimento a una specifica modalità di rappresentazione di grafi, determinare i costi computazionali dell'algoritmo (tempo e spazio). La modalità di rappresentazione scelta sarà oggetto di valutazione.