

Donat un array `a` d'`int` i un enter `m`, escriu un mètode **recursiu** que comprovi si existeix una parella `a[i]` i `a[f]`,  $0 \leq i \leq f < a.length$ , de components *simètriques* (la distància d'`i` a `0` és la mateixa que la distància de `f` a `a.length - 1`) que sumen `m`. Si existeix la parella, ha de tornar l'índex en el que es troba (l'índex `i`, el més baix de la parella), `-1` en cas contrari.

Per exemple, per a `m = 4` i `a = {1, 4, 5, 9, 6, 0, -8}` el mètode en la crida inicial que s'extenga sobre tot l'array ha de tornar `1`, per a `m = 18` i `a = {1, 4, 5, 9, 6, 0, 2}` ha de tornar `3`, per a `m = 25` i `a = {1, 3, 2, 5, 4, 6}` ha de tornar `-1`.

**Es demana:**

- a) Perfil del mètode, amb els paràmetres adequats per tal de resoldre recursivament el problema, i precondició relativa als paràmetres.
- b) Cas base i cas general de la recursió.
- c) Implementació en Java del mètode recursiu.
- d) Crida inicial al mètode recursiu per a comprovar si existeix una parella de components simètriques en un array `dades` que sumen `s`.

El següent mètode **iteratiu** transposa una matriu quadrada:

```
/** Precondició: m és una matriu quadrada */
public static void transposada(int[][] m) {
    for (int i = 0; i < m.length; i++) {
        for (int j = 0; j < i; j++) {
            int aux = m[i][j];
            m[i][j] = m[j][i];
            m[j][i] = aux;
        }
    }
}
```

**Es demana:**

- Indica quina és la talla o grandària del problema, així com l'expressió que la representa.
- Indica, i justifica, si existeixen diferents instàncies significatives per al cost temporal de l'algorisme i identifica-les si és el cas.
- Escull una unitat de mesura per a l'estimació del cost (passos de programa, instrucció crítica) i d'acord amb ella obté una expressió matemàtica, el més precisa possible, del cost temporal del mètode, distingint el cost de les instàncies més significatives en cas d'haver-les.
- Expressa el resultat anterior utilitzant notació asimptòtica.

El següent mètode **recursiu** ordena per *inserció directa* un array  $v[0..pos]$ ,  $0 \leq pos < v.length$ . Si es vol ordenar tot un array **dades**, la crida inicial al mètode seria: **insDirectaRec**(dades, dades.length - 1);

```
/** 0 <= pos < v.length */
public static void insDirectaRec(int[] v, int pos) {
    if (pos > 0) {
        insDirectaRec(v, pos - 1);
        int x = v[pos];
        int j = pos - 1;
        while (j >= 0 && v[j] > x) {
            v[j + 1] = v[j];
            j--;
        }
        v[j + 1] = x;
    }
}
```

**Es demana:**

- Indica quina és la talla o grandària del problema, així com l'expressió que la representa.
- Indica, i justifica, si existeixen diferents instàncies significatives per al cost temporal de l'algorisme i identifica-les si és el cas.
- Escriu l'equació de recurrència del cost temporal en funció de la talla per a cadascun dels casos si hi ha més d'un, o una única equació si únicament hi haguera un cas. Ha de resoldre's per substitució.
- Expressa el resultat anterior utilitzant notació asimptòtica.