

PRG - ETSInf. TEORIA. Curs 2018-19. Parcial 1.
1 d'abril de 2019. Duració: 2 hores.

Nota: L'examen s'avalua sobre 10 punts, però el seu pes específic en la nota final de PRG és **de 3 punts**.

1. 4 punts Donats un array d'enters a i un enter k , $k \geq a.length$, es diu que a és *un capicua de clau k* si la diferència en valor absolut entre les components primera i última és menor que k , menor que $k - 1$ entre les components segona i penúltima, menor que $k - 2$ entre la tercera i l'antepenúltima, i així successivament. S'entén que l'array buit o amb una sola component són capicues, per a qualsevol clau. Es vol un mètode recursiu que comprovi si a és un capicua de clau k . Per exemple,
- per a l'array $\{20, 15, 14, 32, 10, 7, 22\}$ i la clau 10, el mètode ha de retornar **true** ja que:
 - $|20 - 22| < 10$,
 - $|15 - 7| < 9$,
 - $|14 - 10| < 8$ i
 - trivialment, $|32 - 32| < 7$.
 - per a l'array $\{20, 15, 14, 10, 7, 22\}$ i la clau 10, el mètode també ha de retornar **true** ja que:
 - $|20 - 22| < 10$,
 - $|15 - 7| < 9$,
 - $|14 - 10| < 8$.
 - per als arrays anteriors i la clau 9 el mètode ha de retornar **false**, atès que:
 - $|20 - 22| < 9$, però
 - $|15 - 7|$ no és menor que 8.

Es demana:

- a) (0.75 punts) Perfil del mètode recursiu, amb els paràmetres adequats per a resoldre recursivament el problema, i preconditioni relativa a aquests paràmetres.
 - b) (1.25 punts) Cas base i cas general de la recursió.
 - c) (1.50 punts) Implementació en Java del mètode recursiu.
 - d) (0.5 punts) Crida inicial al mètode per a comprovar si un cert array **data** és o no capicua de clau **k**.
2. 3 punts Donada una matriu quadrada a de reals, el següent mètode comprova si la suma dels elements que estan per davall de la diagonal principal menys la dels que estan per damunt, coincideix amb la suma dels elements de la diagonal principal.

```
/** Precondició: a és una matriu quadrada. */
public static boolean sumBelowAbove(double[][] a) {
    double sumBelow = 0, sum = 0, sumAbove = 0;
    for (int i = 0; i < a.length; i++) {
        for (int j = 0; j < a.length; j++) {
            if (j < i) { sumBelow += a[i][j]; }
            else if (j == i) { sum += a[i][i]; }
            else { sumAbove += a[i][j]; }
        }
    }
    return sumBelow - sumAbove == sum;
}
```

Es demana:

- a) (0.25 punts) Indica quina és la grandària o talla del problema, així com l'expressió que la representa.
- b) (0.75 punts) Indica, i justifica, si existeixen diferents instàncies significatives per al cost temporal de l'algorisme i identifica-les si és el cas.
- c) (1.50 punts) Tria una unitat de mesura per a l'estimació del cost (passos de programa, instrucció crítica) i d'acord amb ella obtingues 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.
- d) (0.50 punts) Expressa el resultat anterior utilitzant notació asimptòtica.

3. 3 punts Una *matriu triangular superior* és una matriu quadrada els valors de la qual per davall de la diagonal principal són tots iguals a 0. El mètode recursiu `isUpperTriangular` següent comprova si les files de la matriu `m`, des de 0 fins a `nRow` compleixen aquesta propietat. Així, per a comprovar si certa matriu `mat` és triangular superior faríem la crida `isUpperTriangular(mat, mat.length - 1)`.

```
/** Precondició: m és una matriu quadrada d'enters, -1 <= nRow < m.length */
public static boolean isUpperTriangular(int[][] m, int nRow) {
    boolean res = true;
    if (nRow >= 0) {
        int j = 0;
        while (j < nRow && res) {
            if (m[nRow][j] != 0) { res = false; }
            else { j++; }
        }
        if (res) { res = isUpperTriangular(m, nRow - 1); }
    }
    return res;
}
```

Es demana:

- a) (0.25 punts) Indica quina és la grandària o talla del problema, així com l'expressió que la representa.
- b) (0.75 punts) Indica, i justifica, si existeixen diferents instàncies significatives per al cost temporal de l'algorisme i identifica-les si és el cas.
- c) (1.50 punts) Escriu l'equació de recurrència del cost temporal en funció de la talla per a cadascun dels casos si hi haguera varis, o una única equació si només hi haguera un cas. Resol-la(es) per substitució.
- d) (0.50 punts) Expressa la funció de cost utilitzant notació asimptòtica.