FI2 (12 CFU, annuale), ASD Nettuno (6 CFU), ASD (5 CFU), ASD Nettuno (3 CFU, risolvere due problemi a scelta) - Compito di esame del 19-2-2013 – a.a. 2012-13

tempo a disposizione: 90 minuti

Problema 1

Si considerino i seguenti metodi Java

```
static void recurse(char[] s, int[] p) { // assumere p.length == s.length
  recurse(s, p, 0, s.length-1);
}

static void recurse(char[] s, int[] p, int i, int j) {
  if(i == j) p[i] = 1;
  else {
    int k = (i + j) / 2;
    recurse(s, p, i, k);
    recurse(s, p, k+1, j);
    join(s, p, i, k, j);
  }
}

static void join(char[] s, int[] p, int i1, int i2, int i3) {
  for(int i = i1; i <= i2; i++)
    for(int j = i2+1; j <= i3; j++)
      if(s[i] == s[j]) {
        p[i]++; p[j]++;
      }
}
```

(a) Calcolare, motivandolo adeguatamente, il costo computazionale del metodo `recurse(char[], int[])`, in funzione della dimensione dell'input.

(b) Quale sarà il contenuto di `p` dopo l'esecuzione di `recurse` con input `s = "mamma"` (la notazione è semplificata, in quanto si dovrebbe descrivere `s` come array di `char` e non come stringa) e `p` arbitrario, ma della stessa dimensione di `s`?

(c) Descrivere un algoritmo asintoticamente più efficiente (in termini di tempo) che calcoli la stessa funzione di `recurse`.

Problema 2

Con riferimento ad alberi binari di ricerca i cui nodi contengono (solo) chiavi `int`, risolvere i seguenti punti:

(a) Definire due classi Java, `BST` e `BSTNode`, per rappresentare alberi della tipologia specificata. Nelle classi specificare le variabili membro e le firme di costruttori e metodi.

(b) Definire un metodo Java `List<Integer> notInRange(int a, int b)` della classe `BST` che crea e restituisce una `List` contenente tutte le chiavi di `this` BST esterne all'intervallo $[\texttt{a}, \texttt{b}]$ (restituire `null` se il BST non contiene chiavi esterne all'intervallo). Determinare il costo computazionale del metodo.

(c) Definire un metodo Java `int maxGap()` della classe `BST` che restituisce la max distanza fra due chiavi consecutive. In altre parole, denominate con $x_1, x_2, \ldots, x_n$ le chiavi dell'albero e con $x_{i_1}, x_{i_2}, \ldots, x_{i_n}$ un loro ordinamento in senso crescente, il metodo `maxGap` restituisce $\max_{1 \leq j < n} x_{i_{j+1}} - x_{i_j}$. Risolvere il punto *senza* ordinare le chiavi e determinare il costo computazionale del metodo.

Problema 3

(a)

(b)

(c)