FI2 (12 CFU), ASD Nettuno (6 CFU) - Compito di esame del 27-4-2012 – a.a. 2011-12, *tempo a disposizione: 90 minuti*

Problema 1

Si considerino i seguenti metodi:

```
static void swap(StringBuffer s, int i, int j) {
  char tmp = s.charAt(i);
  s.setCharAt(i, s.charAt(j));
  s.setCharAt(j, tmp);
}

static int m3(StringBuffer s, int k, String[] o, int l) {
  if(k >= s.length()) {
    o[l] = new String(s);
    return l+1;
  } else {
    for(int i = k; i < s.length(); i++) {
      swap(s, k, i);
      l = m3(s, k+1, o, l);
      swap(s, k, i);
    }
    return l;
  }
}

static String[] m2(int n) {
  int s = 1;
  while(n > 0) {
    s *= n;
    n--;
  }
  return new String[s]; // assumere O(1)
}

static String[] m1(StringBuffer s) {
  String[] o = m2(s.length());
  m3(s, 0, o, 0);
  return o;
}
```

(a) Calcolare, motivandolo adeguatamente, il costo computazionale del metodo `m2(int)` in funzione della dimensione dell'input.

(b) Calcolare, motivandolo adeguatamente, il costo computazionale del metodo `m1(StringBuffer)` in funzione della dimensione dell'input.

(c) Qual è il contenuto dell'array restituito da `m1` quando questo viene chiamato sull'input `"abc"`?

Problema 2

Definire una opportuna struttura ad albero per la rappresentazione semplificata di un file system, che può contenere cartelle, file e link (simbolici, cioè espressi come percorsi assoluti). Per ciascuna cartella si desidera rappresentare il nome e il numero di elementi contenuti; per ciascun file, il nome e la dimensione in byte; per ciascun link, il nome e il percorso assoluto di un elemento del file system a cui il link fa riferimento (non deve esserne garantita l'esistenza). All'interno di una stessa cartella non possono coesistere due elementi con lo stesso nome.

(a) Definire la classe Java `FileSystem` e le eventuali altri classi a supporto per la rappresentazione semplificata di file system e la soluzione dei punti successivi (per ciascuna classe: campi, costruttori e firme dei metodi).

(b) Definire e dettagliare il metodo `search(String name)` della classe `FileSystem` che, data un stringa `name`, restituisce una opportuna `List<E>` i cui elementi contengono i percorsi assoluti (stringhe) di file o cartelle (sono esclusi i link) di nome `name`. Qualora non esistano occorrenze di `name`, restituire `null`.

(c) Definire e dettagliare il metodo `long getSize(.)` della classe `FileSystem` che, dato un riferimento a un elemento del file system (file o cartella), restituisce la dimensione dell'elemento. In particolare, nel caso di cartella, si dovrà restituire la somma delle dimensioni in byte dei file contenuti nella porzione di file system la cui radice è la cartella stessa (senza seguire i link simbolici).

Facoltativo: scrivere una variante del metodo `getSize` che, nel calcolo della somma delle dimensioni, segua anche i link simbolici.

Problema 3

(a) Definire una struttura dati concreta per la gestione di insiemi disgiunti, che supporti le operazioni di `find`, `makeSet` e `union`. Descrivere i tre algoritmi.

(b) Discutere le euristiche per migliorare le prestazioni della struttura dati di cui al punto precedente.

(c) Illustrare un algoritmo per il calcolo di minimum spanning tree che sfrutti la struttura dati del punto (a).