ESAME 25 OTTOBRE 2022

Esercizio 1 (10 punti):
Si consideri la seguente funzione:

```
def es1(n):
    if n <= 1:  return 5          Θ(1) CASO BASE
    a = es1(n//2)                 T(n/2)
    i = j = 1
    while j < n:                  log_2(n) VOLTE
        j *= 2
        i += 1                    i = log(n)   J = n
    u, j = 1, n
    while j > 1:                  n/log(n) VOLTE
        j -= i
        u += 1
    return a + es1(n//2) + u      T(n/2)
```

a) Si imposti la relazione di ricorrenza che ne definisce il tempo di esecuzione giustificando dettagliatamente l'equazione ottenuta.

b) Qualora sia possibile, risolvere la ricorrenza utilizzando il **teorema principale** dettagliando il caso del teorema ed i passaggi logici. Se il teorema principale non è applicabile spiegarne il motivo.

$$T(n) = 2T\left(\frac{n}{2}\right) + \Theta(\log(n)) + \Theta\left(\frac{n}{\log(n)}\right)$$

$$T(n) = 2T\left(\frac{n}{2}\right) + \Theta\left(\frac{n}{\log(n)}\right)$$

$$T(1) = \Theta(1)$$

METODO PRINCIPALE

$$n^{\log_b a} = n$$

$$f(n) = \frac{n}{\log(n)}$$

$$\exists\, \varepsilon > 0 \text{ t.c. } \Theta\left(\frac{n}{\log(n)}\right) = O\left(n^{1-\varepsilon}\right)\ ?$$

NON SI PUO' APPLICARE

METODO ITERATIVO

$$T(n) = 2T\left(\frac{n}{2}\right) + \Theta\left(\frac{n}{\log(n)}\right) = 2\left[2T\left(\frac{n}{2^2}\right) + \Theta\left(\frac{n/2}{\log(\frac{n}{2})}\right)\right] + \Theta\left(\frac{n}{\log(n)}\right)$$

$$= 2^k T\left(\frac{n}{2^k}\right) + \sum 2^i \Theta\left(\frac{n}{2^i \log(\frac{n}{2^i})}\right) = \Theta(n) + n \sum^{\log(n)} \Theta\left(\frac{1}{\log(n/2^i)}\right)$$

$$= \Theta(n) + n \sum \frac{1}{\log(n) - i} = \Theta(n) + n \sum_{j=1}^{\log(n)} \frac{1}{j} = \Theta(n \log(\log(n)))$$

Esercizio 2 (10 punti):

Sia A un array di n elementi con n pari, contenente uno stesso numero di interi pari ed interi dispari. Un *riarrangiamento* degli interi in A è *valido* se nelle posizioni ad indice pari compaiono interi pari e in quelle ad indice dispari interi dispari (l'indice 0 è considerato pari).

Ad esempio, per $A = [7, 3, 1, 8, 8, 2, 1, 4]$ esistono diversi riarrangiamenti validi come: $[8, 7, 2, 3, 8, 1, 4, 1]$, oppure $[4, 1, 2, 1, 8, 3, 8, 7]$ o anche $[2, 3, 8, 1, 8, 7, 4, 1]$.

Progettare un algoritmo che, preso l'array A, produca un riarrangiamento valido.

L'algoritmo deve avere costo computazionale $O(n)$ e deve utilizzare uno spazio di lavoro costante (in altri termini, non è possibile utilizzare liste concatenate o array di appoggio).

Dell'algoritmo proposto:

a) si dia la descrizione a parole,

b) si scriva lo pseudocodice,

c) si giustifichi il costo computazionale.

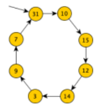
INIZIALIZZO DUE INDICI, $i = 0$ $J = 1$, OGNI VOLTA CHE TROVO UN PARI LO SCAMBIO CON i ED INCREMENTO DI 2.

```
DEF ES3(A):
    i = 0;                                  Θ(1)
    J = 1;                                  Θ(1)
    K = 0;                                  Θ(1)
    WHILE(K < LEN(A)):                      n VOLTE
        IF (A[K] % 2 == 0):                 Θ(1)
            A[K], A[i] = A[i], A[K]         Θ(1)
            i += 2;                         Θ(1)
        K += 1;                             Θ(1)
    RETURN A;                               O(1)
```

$$\Theta(1) + n\Theta(1) = \Theta(n)$$

Esercizio 3 (10 punti):
Si consideri una lista a puntatori circolare L data tramite un puntatore p ad un suo elemento. In L ogni nodo ha 2 campi: il campo `key` contenente un intero ed il campo `next` con il puntatore al nodo seguente.

Sappiamo che gli interi dei vari nodi sono tutti distinti e bisogna trovare il valore minimo tra questi. Ad esempio per la lista circolare di seguito il valore cercato è 3:



Progettare un algoritmo **iterativo** che, dato il puntatore p ad un nodo della lista circolare, restituisce il valore cercato in tempo $\Theta(n)$ dove n è il numero di nodi della lista.
Dell'algoritmo proposto:

a) si dia la descrizione a parole,

b) si scriva lo pseudocodice,

c) si giustifichi il costo computazionale.

SCORRO LA LISTA, OGNI VOLTA CHE TROVO UN VALORE PIÙ PICCOLO DEL MINIMO LO AGGIORNO, SE TROVO UN VALORE UGUALE AL MINIMO, LO RITORNO

```
DEF ES3(P):
    MIN = P.KEY ;
    WHILE(TRUE):
        IF (P.NEXT.KEY ==MIN):
            RETURN P.NEXT.KEY):
        IF (P.KEY < MIN):
            MIN = P.KEY;
        P=P.NEXT;
```