Esercizio 1 (10 punti): Per la soluzione di un certo problema disponiamo di un algoritmo iterativo con costo computazionale $\Theta(n^3)$. Ci viene proposto in alternativa un algoritmo ricorsivo il cui costo è catturato dalla seguente ricorrenza:

$T(n) = a \cdot T\left(\frac{n}{2}\right) + \Theta(\sqrt{n})$ per $n \geq 2$

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

dove a è una certa costante intera positiva con $a \geq 2$.

Determinare quale sia il valore massimo che la costante intera a può avere perché l'algoritmo ricorsivo risulti asintoticamente più efficiente dell'algoritmo iterativo di cui disponiamo. **Motivare bene la risposta.**

PRIMA DI TUTTO, VEDIAMO QUANTO COSTEREBBE L'ALGORITMO RICORSIVO CON IL VALORE GENERICO a.

metodo principale

$n^{\log_b a} = n^{\log_2 a}$ ESSENDO $a \geq 2$,

$f(n) = n^{\frac{1}{2}}$ $\log_2 a \geq 1$, QUINDI

$$f(n) = O(n^{\log_2 a - \varepsilon})$$

QUINDI $T(n) = n^{\log_2 a}$

ORA MI CHIEDO, QUAL'È IL VALORE MASSIMO DI a, TALE CHE $n^{\log_2 a} < n^3$?

CHIARAMENTE: $\log_2 8 = 3$, QUINDI IL VALORE MASSIMO CHE PUÒ ASSUMERE a È 7.

Esercizio 2 (10 punti):
Dati due arrays A e B, rispettivamente di n ed m interi distinti, con $m < n$, si vuole sapere se l'array A contenga l'array B come sottoarray.
Ad esempio, se $A = [5, 9, 1, 3, 4, 8, 2]$, per $B = [3, 4, 8]$ la risposta è *SI* mentre per $B = [3, 8, 2]$ o $B = [9, 6, 8]$ la risposta è *NO*.

Progettare un algoritmo che, dati gli arrays A e B, restituisca 1 se la risposta al problema è *SI*, 0 altrimenti. Il costo computazionale dell'algoritmo deve essere $O(n)$.
Dell'algoritmo proposto:

a) si dia la descrizione a parole,
b) si scriva lo pseudocodice,
c) si giustifichi il costo computazionale.

SEMPLICEMENTE, "CAMMINERÒ" SULL'ARRAY A, ED OGNI VOLTA CHE IL VALORE VISITATO SARÀ UGUALE AL CORRENTE VALORE DI B, CAMMINERÒ SU B, FINCHÉ I VALORI SARANNO UGUALI, SE NE TROVERÒ UNO DIVERGENTE, L'INDICE CHE CAMMINAVA SU B TORNERÀ A 0, SE FINIRÒ DI SCORRERE B, RITORNERÒ 1. ALTRIMENTI 0.

```
DEF ES2(A, B):
    i, J = 0;                       Θ(1)
    WHILE (i < LEN(A)):             n VOLTE
        IF (J = LEN(B-1)):          Θ(1)
            RETURN 1;               Θ(1)
        IF (A[i] == A[J]):          Θ(1)
            J += 1;                 Θ(1)
        ELSE:                       Θ(1)
            J = 0;                  Θ(1)
        i += 1;                     Θ(1)
    RETURN 0;                       Θ(1)
```

$$T(n) = 2\Theta(1) + n[5\Theta(1)] = \Theta(1) + \Theta(n) = \Theta(n)$$

CASO PEGGIORE $\Theta(n)$

CASO MIGLIORE $O(n)$

Esercizio 3 (10 punti): Sia dato un albero binario T, in cui ogni nodo p ha tre campi: il campo valore $p.val$, il campo col puntatore al figlio sinistro $p.sx$ e il campo col puntatore al figlio destro $p.dx$, in mancanza di figlio il puntatore vale $None$.
Progettare un algoritmo *ricorsivo* che, dato il puntatore p alla radice dell'albero binario T, restituisca 1 se tutti i nodi dell'albero hanno lo stesso valore, 0 altrimenti. Il costo computazionale dell'algoritmo deve essere $O(n)$, dove n è il numero di nodi dell'albero.
Dell'algoritmo proposto:

a) si dia la descrizione a parole,

b) si scriva lo pseudocodice,

c) si giustifichi il costo computazionale.

BANALMENTE, CONTROLLO OGNI NODO, ALLA RADICE IMPOSTO IL VALORE CHE DOVRANNO ASSUMERE LE CHIAVI DI TUTTI I NODI, DENOMINATO VALUE. CONTROLLERO' OGNI NODO, SE UNO SOLO DI QUESTI AVRA' LA CHIAVE DIVERSA DA VALUE, RITORNERO' 0.

```
DEF ES3(P, VALUE):
    IF(VALUE == NONE):
        VALUE = P.VAL;  // RADICE
    RESULT = P.VAL == VALUE;
    IF(!RESULT): RETURN 0 ;
    IF(P.SX):
        RESULT = RESULT AND ES3(P.SX, VALUE) == 1;
    IF(P.DX):
        RESULT = RESULT AND ES3(P.DX, VALUE) == 1;
    IF (RESULT): RETURN 1;
    RETURN 0;
```

OGNI VISITA HA TEMPO COSTANTE. DEFINIAMO K I FIGLI SINISTRI:

$$T(n) = T(K) + T(n-K-1) + \Theta(1) \qquad T(1) = \Theta(1)$$

RISOLVO CON METODO DI SOSTITUZIONE

$$T(n) = T(K) + T(n-K-1) + c \qquad T(1) = d$$

IPOTIZZO $T(n) = \Theta(n) \rightarrow an \leq T(n) \leq bn$

CASO BASE:

$a \leq d \leq b$

IPOTESI INDUTTIVA

$\forall m < n \rightarrow T(m) = \Theta(m) \Rightarrow am \leq T(m) \leq bm$

PASSO INDUTTIVO

$$T(K) + T(n-K-1) + c \geq an \qquad T(K) + T(n-K-1) + c \leq bn$$

essendo $K < n$, SI HA L'IPOTESI INDUTTIVA

$$aK + (n-K-1)a + c \geq an \qquad bK + (n-K-1)b + c \leq bn$$

$$\cancel{aK} + an - \cancel{aK} - a + c \geq an \qquad \cancel{bK} + bn - \cancel{bK} - b + c \leq bn$$

$$an - a + c \geq an \qquad bn - b + c \leq bn$$

$$c \geq a \qquad c \leq b$$

VERIFICATO