Esame 8/9/2021 - Prima parte

Punti: 6/10

1

(0/1 punto)

Sia **a** una stringa, quale tra le chiamate alla funzione enigma restituisce sempre **Tr**

```
def enigma(x):
    if x > 100 or x < 0:
        return [1]
    else:
        return [1]+enigma(x+1)
```

- enigma(a+a[::-1]) ✓
- enigma(2*a)
- enigma(a[:2])

2

(1/1 punto)

La funzione **binsearch_iter** implementa l'algoritmo di ricerca binaria in modo ite
la funzione **binsearch_ric** implementa lo stesso algoritmo in modo ricorsivo. Qu
seguenti affermazioni è sempre vera?

- binsearch_ric è più efficiente rispetto al tempo
- le due funzioni hanno lo stesso costo rispetto alla memoria supplementare utilizzata
- binsearch_iter è più efficiente dal punto di vista della memoria supplementare utilizzata ✓
- binsearch_iter è più efficiente rispetto al tempo

3

(1/1 punto)

Sia **A** un array dinamico di capacità c e contenente $n = c$ elementi. Su **A** si esegu
operazioni alternate di inserimento e cancellazione (la prima è un inserimento). Qu
ridimensionato **A**?

- Mai
- Una volta ✓
- $2n$ volte
- n volte

4

(0/1 punto)

Sia **n** un intero > 1069, quanto vale **c** al termine del seguente frammento di codic

```
c = 0
for i in range(n):
    for j in range(n-i):
        c+=1
```

- n*n
- n*n/2
- n(n+1)/2 ✓
- n(n-1)/2

5

(1/1 punto)

Sia **a** un array di n float tutti compresi tra 0 e 1, qual è il costo dell'algoritmo miglio
a?

- quadratico in n
- ordine di $n\log(n)$ ✓
- ordine di $\log(n)$
- lineare in n

6

(0/1 punto)

Si consideri la seguente funzione C:

```
int f(char *x){
    int i;
    for( i = 0; i < strlen(x); i++ )
        printf("%s\n", x+i);
}
```

Sia **a** una stringa di lunghezza n, qual è il tempo di esecuzione di `f(a)`?

- Lineare in n
- Quadratico in n ✓
- Cubico in n
- Costante

7

(1/1 punto)

Sia **a** una stringa di lunghezza n, **d** un dizionario di dimensione m e **s** la struttura d seguente codice.

```
s = set()
for c in a:
    s.add(c)
for k in d:
    s.add(k)
```

Dire quale tra le seguenti affermazioni è sempre vera.

- len(s) $<n+m$
- len(s) $\leq n+m$ ✓
- len(s) $>\max(n,m)$

len(s) $=n+m$

8

(0/1 punto)

Siano **v** e **p** due interi positivi con **p < v**, cosa restituisce **f(v,p)**?

```
int f(int a, int p){
    int i;
    int **b = malloc(a*sizeof(int*));
    b[0] = &a;
    for(i = 1; i < a; i++){
        b[i] = b[i-1];
    }
    a = p;
    return *b[p];
}
```

- 0
- il valore di p ✓
- il valore di v
- un indirizzo

9

(1/1 punto)

Sia a una lista di n interi, qual è il costo computazionale di `f(a)` in funzione di n?

```
def f(a):
    n = len(a)
    if n >= 2:
        f(a[:int(n/2)])
```

- Tempo lineare in n, memoria costante.
- Tempo e memoria lineari in n.
- Tempo e memoria ordine di $\log(n)$. ✓
- Tempo ordine di $\log(n)$, memoria costante.

10

(1/1 punto)

Sia **a** una lista di n numeri. La funzione **booh_min(a)** restituisce la coppia **(m, p** minimo in **a** e **p** è la sua posizione. Qual è il costo di **booh_min(a)** in termini di te

```
def booh_min(a):
    n = len(a)
    for i in range(n):
        if a[i] == min(a[i:]):
            return a[i], i
```

- Ordine di n nel caso peggiore, costante nel caso migliore
- Sempre ordine di $n*n$
- Ordine di $n*n$ nel caso peggiore, ordine di n nel caso migliore ✓
- Sempre ordine di n.

Torna alla pagina di ringraziamento