Esame 6/7/2020

Punti: 10/10

1

(1/1 punti)

Siano a ed f definite come segue, qual è il tipo di a[0] dopo l'esecuzione di

```
a.sort(key=f)
```

?

```
a = [100, [8,'quattro'], 6, (3,11,28), 9]
```

```
def f(e):
    if type(e) != type(0):
        return e[0]
    else:
        return e
```

- ○ Lista
- ◉ Tupla ✓
- ○ Intero

2

(1/1 punti)

Sia n un intero maggiore di 6, qual è il risultato di `enigma(n)`?

```
def enigma(n):
    a = list(range(n))
    A = set(a)
    if len(A) > 0:
        B = set(a*6)
        return 1+enigma(len(B-A))
    else:
        return 0
```

- n
- 1 ✓
- 0
- 6

3

(1/1 punti)

a e b sono due liste concatenate contenenti interi e d un dizionario, inizialmente vuoto, implementato con liste di trabocco. Gli elementi di d sono coppie `(k, v)` dove la chiave k è di tipo intero e la chiave v è di tipo puntatore. Vengono eseguite le seguenti operazioni:

- per ogni elemento x di a, la coppia (x, NULL) viene inserita in d;
- per ogni elemento x di b, la coppia (x, NULL) viene inserita in d.

Se a contienene tutti e soli gli elementi di un insieme A e b contienene tutti e soli gli elementi di un insieme B, quanti elementi contiene d?

- $|A \cup B|$ ✓
- $|A \cap B|$
- $|A - B|$
- $|A| + |B|$

4

(1/1 punti)

Se a è un intero e b una stringa cosa restituisce `maximum(a, b)`?

```
def maximum( x, y ):
    if x > len(y):
        mx = x
    else:
        mx = y
    return mx
```

- ○ Un intero
- ○ Una stringa
- ◉ Un numero o una stringa ✓

5

(1/1 punti)

Sia n un intero positivo maggiore di 1000 e `a = [1]*n`. Qual è la lunghezza di a dopo l'esecuzione di `remitems(a)`?

```
def remitems( a ):
    i = 1
    while i < len(a):
        if a[i] in a[:i]:
            del a[i]
        i += 1
```

- n
- circa n/2 ✓
- 1
- 0

6

(1/1 punti)

Sia a una stringa di lunghezza n > 1, qual è la lunghezza di f(a) dove f() è la funzione C così definita?

```
char *f(char *a){
    int i, n = strlen(a);
    char *b = malloc(n+1);
    for (i = 0; i < n/2; i++){
        b[i] = a[n-i];
    }
    return b;
}
```

- n
- n+1
- 0 ✓
- Circa n/2

7

(1/1 punti)

Qual è il costo computazionale della seguente funzione C in termini di memoria supplementare utilizzata?

```c
int f(int n){
    int i, j;
    int *a = malloc(sizeof(int)*n);
    int *b;
    for(i = 1; i < n; i++){
        b = malloc(sizeof(int)*2);
        b[0] = i;
        b[1] = i+1;
        a[i] = b[0]+b[1];
    }
    free(a);
    free(b);
    return a[n-1];
}
```

- (●) Lineare in n ✓
- ( ) Quadratica in n
- ( ) Cubica in n
- ( ) Costante

8

(1/1 punti)

Si consideri la seguente funzione C:

```
void f(int *x, int i){
    *(x+x[i]) = 2*(*(x+i));
}
```

Sia a un array di n>0 interi tale che a[i] = i per ogni indice i e sia 0≤k<n. Dopo l'invocazione di f(a,k) qual è il valore di a[k]?

- k
- 2k ✓
- 0
- n

9

(1/1 punti)

a è una lista di n interi ordinati in modo crescente e k un intero. Qual è il costo computazionale dell'algoritmo più efficiente per ordinare `a.append(k)`?

- Quadratico in n
- Ordine di n log(n)
- Lineare in n ✓
- Costante
- Lineare in k

10

(1/1 punti)

Sia a una lista concatenata contenente n interi pari ordinati in modo crescente seguiti da n interi dispari ordinati in modo crescente. Qual è il costo computazionale dell'algoritmo più efficiente per ordinare tutti gli elementi di a su una seconda lista?

- (●) ordine di n ✓
- ( ) ordine di n log (n)
- ( ) quadratico in n
- ( ) costante