

Chi cerca trova

Descrizione

Avete 3 ore di tempo per portare a termine la simulazione.

Di seguito le tracce, solo chi riesce a completare con successo le prime due puo' cimentarsi, se vuole, con la terza, per un'esperienza piu' sfidante.

Considerate che si puo' raggiungere il massimo anche solo con le prime due, per evitare di penalizzare chi non ha mai programmato prima.

Valutazione

Pulizia del codice, pseudocodice, modularita', nomi delle funzioni e delle variabili e tutto cio' che ci siamo detti in classe in queste settimane saranno oggetto di valutazione.

1 - Find and replace

Scrivere una funzione che riceva in ingresso tre stringhe:

- la prima sarà un testo
- la seconda sarà una parola che andrà cercata nel testo
- la terza sarà la parola da sostituire al posto della seconda

La funzione deve quindi produrre lo stesso effetto che si ottiene quando si fa find and replace di una parola in un testo.

Si assuma che:

- la seconda e la terza stringa siano sempre e solo una parola, mai una frase, quindi non ci saranno spazi
- sulle stringhe non esistano i metodi indexOf, replace, e similari, dovete scrivere voi l'algoritmo
- la seconda e la terza parola non necessariamente devono avere lo stesso numero di caratteri

Ricordate che le stringhe in JavaScript sono trattate in modo simile agli array.

La funzione deve ritornare la nuova stringa aggiornata.

2 - Find and update

Scrivere una funzione che, dato in ingresso un array di oggetti così strutturato:

```
1  [  
2    {  
3      "maggiorenne": null,  
4      "type": "boolean"  
5    }, {  
6      "nome": null,  
7      "type": "string"  
8    }, {  
9      "cognome": "Rossi",  
10     "type": "string"  
11   }, ...  
12 ]
```

sia in grado di attribuire un valore di default dove sia presente un null, seguendo queste regole:

- se il type è "boolean" deve aggiornare usando false
- se il type è "string" deve aggiornare usando stringa vuota
- se il type è "number" deve aggiornare usando 0
- se il type è "array" deve aggiornare usando array vuoto
- se il type è "object" deve aggiornare usando oggetto vuoto

Come vedete ogni singolo oggetto ha sempre due attributi, uno la cui chiave non è dato a sapere a priori, un altro la cui chiave è sempre "type" e il valore è nella lista qui sopra.

La funzione deve ritornare lo stesso oggetto ricevuto in input, con i valori aggiornati.

3 - L'Albero del Re

Affrontare questo esercizio solo se si sono risolti i precedenti due.

Dato questa struttura ad albero:

```
1  {
2    "value": "Arwen",
3    "left": {
4      "value": "Eärendil",
5      "left": {
6        "value": "Nimloth the Fair"
7        "left": null,
8        "right": null
9      },
10     "right": {
11       "value": "Galadriel",
12       "left": {
13         "value": "Eowyn",
14         "left": null,
15         "right": null
16       },
17       "right": null
18     }
19   },
20   "right": {
21     "value": "Shelob"
22     "left": null,
23     "right": null
24   }
25 }
```

navigarla visitando i nodi alla ricerca del valore "Nimloth the Fair", se trovato ritornare a che distanza si trova dalla radice dell'albero, se non trovato ritornare -1.

Non e' detto che l'albero con cui valuterò il vostro esercizio avrà questa struttura.

Non abbiamo mai visto questo tipo di strutture dati a lezione, per questo motivo l'esercizio e' interamente opzionale.