

JavaScript: verificare se un albero binario è bilanciato

Un albero binario è una struttura dati composta da nodi, in cui ogni nodo può avere al massimo due figli: un figlio sinistro e un figlio destro. Un albero binario è considerato "bilanciato" quando l'altezza dei suoi sottoalberi destro e sinistro non differisce di più di una unità. Verificare se un albero binario è bilanciato è un problema comune nell'informatica e può essere affrontato utilizzando JavaScript. In questo articolo, esploreremo come farlo.

Per iniziare, è importante comprendere la struttura di un albero binario. Ogni nodo dell'albero è rappresentato da un oggetto che contiene un valore e due proprietà: `left` (sinistra) e `right` (destra), che rappresentano i figli del nodo. Gli alberi binari possono essere rappresentati utilizzando una classe JavaScript personalizzata, ad esempio:

```
class TreeNode {
  constructor(value) {
    this.value = value;
    this.left = null;
    this.right = null;
  }
}
```

Per verificare se un albero binario è bilanciato, dobbiamo calcolare l'altezza dei sottoalberi sinistro e destro di ciascun nodo e verificare che la differenza tra queste altezze sia sempre inferiore o uguale a 1. Possiamo farlo utilizzando una funzione ricorsiva.

Ecco come potrebbe apparire una funzione per verificare la bilanciatura di un albero binario:

```
function isBalanced(root) {
  // Caso base: un albero vuoto è considerato
  bilanciato
  if (!root) {
    return true;
  }

  // Calcola l'altezza dei sottoalberi sinistro e
  destro
  const leftHeight = getHeight(root.left);
  const rightHeight = getHeight(root.right);

  // Verifica se la differenza tra le altezze è <= 1
  if (Math.abs(leftHeight - rightHeight) <= 1) {
    // Continua a verificare i sottoalberi
    return isBalanced(root.left) &&
    isBalanced(root.right);
  }

  // Se la differenza tra le altezze è > 1, l'albero
  non è bilanciato
  return false;
}

function getHeight(node) {
  // Restituisce l'altezza di un sottoalbero
  if (!node) {
    return 0;
  }
  return Math.max(getHeight(node.left),
```

```
getHeight(node.right)) + 1;  
}
```

In questa implementazione, la funzione `isBalanced` verifica se un albero è bilanciato o meno. Per fare ciò, calcola l'altezza dei sottoalberi sinistro e destro utilizzando la funzione `getHeight` e quindi verifica se la differenza tra queste altezze è inferiore o uguale a 1. Se la differenza è maggiore di 1, l'albero non è bilanciato. Altrimenti, la funzione continua a verificare i sottoalberi sinistro e destro.

Ecco un esempio di utilizzo della funzione `isBalanced` con un albero binario:

```
const root = new TreeNode(1);  
root.left = new TreeNode(2);  
root.right = new TreeNode(3);  
root.left.left = new TreeNode(4);  
root.left.right = new TreeNode(5);  
  
console.log(isBalanced(root)); // Restituirà true
```

In questo esempio, l'altezza del sottoalbero sinistro è 2, mentre l'altezza del sottoalbero destro è 1, quindi la differenza è 1, il che rende l'albero bilanciato.

Conclusioni

La verifica della bilanciatura di un albero binario è un problema comune nell'informatica e può essere risolto in JavaScript utilizzando una funzione ricorsiva che calcola l'altezza dei sottoalberi sinistro e destro di ciascun nodo. Speriamo che questo articolo ti abbia aiutato a comprendere come

affrontare questo problema e a implementare una soluzione in JavaScript. La bilanciatura degli alberi binari è importante in molte applicazioni, come le strutture dati degli alberi di ricerca binaria, dove una buona bilanciatura è essenziale per garantire le prestazioni ottimali delle operazioni di ricerca e inserimento.