|  |  |
| --- | --- |
| **Check Max Heap in C++** | |
| #include <iostream>  #include <vector>  using namespace std;  class Solution {  public:  static bool checkMaxHeap(vector<int>& arr) {  for (int i = 0; i < arr.size(); i++) {  int pIndex = i;  int lIndex = 2 \* i + 1;  int rIndex = 2 \* i + 2;  if (lIndex < arr.size() && arr[pIndex] < arr[lIndex]) {  return false;  }  if (rIndex < arr.size() && arr[pIndex] < arr[rIndex]) {  return false;  }  }  return true;  }  };  int main() {  // Example input  vector<int> arr = {42, 20, 18, 6, 14, 11, 9, 4};  // Call the static method checkMaxHeap from Solution class  bool result = Solution::checkMaxHeap(arr);  // Print the result  cout << boolalpha << result << endl;  return 0;  } | Dry Run for Input: {42, 20, 18, 6, 14, 11, 9, 4}  | **Index (i)** | **Parent (arr[i])** | **Left Child Index (2i+1)** | **Left Value** | **Right Child Index (2i+2)** | **Right Value** | **Valid?** | | --- | --- | --- | --- | --- | --- | --- | | 0 | 42 | 1 | 20 | 2 | 18 | ✅ | | 1 | 20 | 3 | 6 | 4 | 14 | ✅ | | 2 | 18 | 5 | 11 | 6 | 9 | ✅ | | 3 | 6 | 7 | 4 | 8 (invalid) | — | ✅ | | 4 to 7 | Leaf nodes | No children | — | — | — | ✅ |   All parent nodes are greater than their children → ✅ **Valid Max Heap** 🖨 Output: true |
| true | |