cs330su21-a.sg

Grade

Reviewed on Friday, 2 July 2021, 2:49 PM by Automatic grade
grade: 100.00 / 100.00

Assessment report [-]
[+]**Summary of tests**

Submitted on Friday, 2 July 2021, 2:49 PM (Download)
q.cpp

```cpp
/*!******************************************************************************
\file    q.cpp
\author  Goh Wei Zhe
\par     DP email: weizhe.goh@digipen.edu
\par     Course: CS330
\par     Section: A
\par     Programming Assignment #1
\date    2-07-2021

\brief Implementation of backtracking algorithm

*******************************************************************************/

#include <iostream>
#include <vector>
#include <numeric> //accumulate

namespace CS330
{
  namespace subsetsum
  {
      bool subset_rec(std::vector<int> const& set, int sum,
      std::vector<int>& subset, unsigned long index)
      {
          int curr_sum = std::accumulate(subset.begin(), subset.end(), 0);

          /*----- Success: find a solution -------*/
          if (curr_sum == sum)
          {
              for (unsigned long i = 0; i < subset.size(); ++i)
              {
                  std::cout << subset[i] << " ";
              }

              std::cout << "  sum " << curr_sum << std::endl;
              return true;
          }

          /*----- Hit the bottom: start backtracking -------*/

          // Add code here
          // Your code should print out the candidate solution before
          // return false
          // The output should be in format (# represents a number):
          // # # # #   sum #

          if (index == set.size())
          {
              if (!subset.empty())
              {
                  for (unsigned long i = 0; i < subset.size(); ++i)
                  {
                      std::cout << subset[i] << " ";
                  }

                  std::cout << "  sum " << curr_sum << std::endl;
                  return false;
              }
          }

          /*----- Backtracking -------*/

          // ADD code here.
          // You backtracking should stop as soon as the first solution is found
          //Backtrack

          // left: add nothing {} to the solution
          subset.push_back(0);

          if (subset_rec(set, sum, subset, index + 1))
              return true;
          else
              subset.pop_back();

          // right: add the element set[index] to the solution
          subset.push_back(set[index]);

          if (subset_rec(set, sum, subset, index + 1))
              return true;
          else
              subset.pop_back();

          return false; // continue backtracking
      }

      std::vector<int> subset_sum(std::vector<int> const& set, int sum)
      {
        std::vector<int> subset;
        CS330::subsetsum::subset_rec(set, sum, subset, 0);
        return subset;
      }
  }
}
```

VPL

◄ Attendance cs330su21-a.sg Thursday 1/07/2021 11:00am-12:40pm

Jump to...

You are logged in as Wei Zhe GOH (Log out)