Sprawozdanie 2

Zadanie 10. Liczbę naturalną C można przedstawić jako sumę parami różnych liczb naturalnych. Na przykład jeśli $C = 6$, to możemy C przedstawić na cztery sposoby:
$1 + 2 + 3$
$1 + 5$
$2 + 4$
6
a jeśli $C = 10$, to takimi podziałami są:
$1 + 2 + 3 + 4$
$1 + 2 + 7$
$1 + 3 + 6$
$1 + 4 + 5$
$1 + 9$
$2 + 3 + 5$
$2 + 8$
$3 + 7$
$4 + 6$
10
Skonstruuj algorytm wyczerpujący z nawrotami, generujący wszystkie podziały podanej liczby naturalnej C.

Adrian Rupala

7 maja 2018

Spis treści

1. Teoria

Algorytm z nawrotami (backtracking) - algorytm wyszukiwania wszystkich lub kilku rozwiązań polegający na znajdowaniu wyniku metodą "prób i błędów", wszelako z oznaczeniem niepowodzeń, dzięki czemu te same błędy nie są popełniane dwukrotnie.

Jeżeli problem pozwala na zastosowanie algorytmu wyszukiwania z nawrotami, to metoda ta może być znaczenie efektywniejsza niż wyszukiwanie wyczerpujące (zakładając przeszukiwanie wszystkich rozwiązań), ponieważ pojedynczy test może wyeliminować nie jedno a wiele rozwiązań niedopuszczalnych.

Rekurencja - technika programowania, dzięki której funkcja, procedura lub podprogram jest w stanie w swoim ciele wywołać sam siebie. Pozwala ona łatwo wykonać wiele zadań, w których potrzeba jest wyników cząstkowych do obliczenia całości.

2. Rozwiązanie

Oto pseudokod przedstawiający rozwiązanie problemu przedstawionego w zadaniu.

```
bool znajdz_duplikat(int lista[], int rozmiar_tablicy) {
   sort(lista);
   for (int i = 0; i < rozmiar_tablicy - 1; i++) {
      if (lista[i] == lista[i + 1]){
         return true;
      }
   }
   return false;
}

void sprawdz_i_wypisz(int pozycja, int pozostalo) {
   if (pozostalo == 0 && znajdzDuplikat(lista, pozycja) == false) {
      for (int i = 1; i <= pozycja - 1; i++) {
         cout << lista[i] << " ";
      }
      cout << endl;
   } else {
      for (int k = lista[pozycja - 1]; k <= pozostalo; k++) {
         lista[pozycja] = k;
         sprawdz_i_wypisz(pozycja + 1, pozostalo - k);
      }
   }
}

void wywolanie(int C) {
   lista[0] = 1;
   sprawdz_i_wypisz(1, C);
}
```