



Programmierpraktikum Technische Informatik (C++)

Aufgabe 05

Hinweise

Abgabe: Stand des Git-Repositories am 25.5.2023 um 9 Uhr.

Die Dateien zur Bearbeitung dieser Aufgabe erhalten Sie, indem Sie die neue Aufgabe aus dem Aufgabenrepository in Ihr lokales mergen. Dies geschieht mit `git pull common main` innerhalb Ihres Repositories. Öffnen Sie ein Terminal in dem Verzeichnis der jeweiligen Aufgabe und geben Sie `code .` ein. Dann öffnet sich VSCode, in dem das CMake-Tools-Plugin installiert sein sollte. Sie können dann in der unteren blauen Zeile die Schaltflächen zum Kompilieren (Build), Starten und ggf. Testen (Run CTest) verwenden. Es sind nicht zu allen Aufgaben CTests vorgesehen. Die Lösungen committen Sie bitte in Ihr lokales Repository und pushen sie in Ihr Repository auf den Gitlab-Server.

Im Folgenden soll eine Floorplanning-Programm erstellt werden. Ein solches Programm wird zum Beispiel benutzt, um Module auf einem Chip möglichst platzsparend anzuordnen. Zur Vereinfachung werden wir davon ausgehen, dass es egal ist, wie weit diese Bauteile voneinander entfernt sind und konzentrieren uns nur auf die Gesamtfläche. Die einzelnen Modul-Zellen (Cell) werden in einem binären Schnittbaum (slicing tree) gespeichert (s.a. <http://edascript.ims.uni-hannover.de/de/4-03-floorplanning/cut-tree.html>). Jeder Blatt-Knoten in diesem Baum steht für eine Zelle (Teilfläche eines Moduls). Jeder innere Knoten steht für eine Schnittoperation (horizontal bzw. vertikal). Die Schnittoperation beschreibt, wie die beiden Teilbäume, die seine Kinder sind, zueinander angeordnet sind: Ein horizontaler Schnitt bedeutet, dass die Teilbäume nebeneinander und ein vertikaler Schnitt bedeutet, dass sie untereinander angeordnet werden sollen.

Teilaufgabe 1 (1.5 Punkte)

Die Funktion `chooseRandomOp` in der Datei `floor.cpp` soll eine zufällige Schnittoperation (`Op::Vertical` oder `Op::Horizontal`) zurückgeben. Der Wert `Op::None` bedeutet dabei keine definierte Schnittoperation. Dieser Wert soll von Ihnen nach der Implementierung nicht mehr zurückgegeben werden. Initialisieren Sie den Zufallsgenerator nur beim ersten Aufruf der Funktion.

Hinweis: Verwenden Sie `std::uniform_int_distribution<T>()`.



Teilaufgabe 2 (0.5 Punkte)

Die Funktion `switchOp` in der Datei `floor.cpp` soll eine ihr übergebene Operation umtauschen. `Op::Horizontal` wird zu `Op::Vertical` und umgekehrt.

Teilaufgabe 3 (2 Punkte)

Im Konstruktor `Floor::Floor` soll der Schnittbaum aus einer übergebenen Liste von Zellen erstellt werden. Füllen Sie zunächst den klasseninternen Vektor für die Zell-Knoten `cellNodes`. Erzeugen Sie dann die inneren Knoten im Vektor `innerNodes` mit zufälliger Schnittrichtung. Die Baumstruktur muss nicht zufällig sein, sondern darf systematisch erzeugt werden. Wichtig ist nur, dass es sich um einen gültigen binären Baum handelt und der Wurzelknoten mit `setRoot` gesetzt wird.

Teilaufgabe 4 (1.5 Punkte)

Die Methode `Floor::calcSize` berechnet die Ausmaße der Teilfläche eines Baums mit seinen Zellen hat. Angefangen beim übergebenen Knoten soll:

- Die Größe des Knotens als Tupel zurückgegeben werden, wenn dieser eine Zelle enthält.
- `Floor::calcVerticalSize` aufgerufen werden, wenn es sich um eine vertikale Schnitt-Operation handelt.
- `Floor::calcHorizontalSize` aufgerufen werden, wenn es sich um eine horizontale Schnitt-Operation handelt.

Die beiden Hilfsfunktionen rufen für beide Kindknoten rekursiv `Floor::calcSize` auf. Die Rückgabewerte müssen dann entsprechend der Operation miteinander verknüpft werden. Bei einem vertikalen Schnitt enthält das Ergebnistupel die Summe der Breiten und das Maximum der Höhen, entsprechend umgedreht für horizontale Schnitte. Implementieren bzw. ergänzen Sie die Methoden `Floor::calcSize`, `Floor::calcVerticalSize` und `Floor::calcHorizontalSize` entsprechend.

Teilaufgabe 5 (1.5 Punkte)

Implementieren Sie die Methode `Floor::modify`. Sie soll an einem zufälligen Knoten eine zufällige Änderung vornehmen. Dabei soll jeweils eine der folgenden Operationen ausgeführt werden:

- Das Tauschen der Operation in einem zufälligen inneren Knoten



- Das Rotieren der Zelle in einem zufälligen Zell-Knoten um 90 Grad.
- Das Austauschen der Kindknoten von einem zufälligen inneren Knoten.
- Shiftoperation an einem inneren Knoten nach links (`Floor::shiftLeft`) oder rechts (`Floor::shiftRight`).

Teilaufgabe 6 (2 Punkte)

Implementieren Sie die Methode `Floor::optimize`. Diese soll iterativ zufällige Veränderungsoperationen des Schnittbaums mit Hilfe von `modify()` ausführen und den sich jeweils ergebenden Floorplan bewerten. Änderungen am Floorplan sollen nicht rückgängig gemacht werden, auch wenn sie eine Verschlechterung ergeben. Der `Floor::optimize` übergebene Parameter stellt die Anzahl der Iterationen dar, die durchgeführt werden sollen. Sollte sich nach einer Veränderung ein Plan mit einer Fläche ergeben, die kleiner ist als die bisher beste Lösung, dann wird sie durch den neuen Plan ersetzt. Den Plan, der durch den aktuellen Schnittbaum definiert ist, können Sie mit der Methode `Floor::pack` im Attribut `Floor::plan` erzeugen.

Hinweis: Wenn die Ausführung zu lange dauert, überprüfen Sie, ob `Floor::packing` nur ausgeführt wird, wenn dies auch notwendig ist.