Project Five: Texture Packing

Date: 2020-05-02

# Chapter 1: Introduction

### Problem Description

Texture Packing is a strip-packing problem and is a common topic explored in the area of Approximation Algorithms. Unlike the *Bin Packing problem*, there are two parameters (width and height) instead of one (bin capacity) are taken into calculation.

Given a set of rectangles with dimensions width and height where , we are expected to pack them into a larger shape with a pre-specified width . We want to pack as many rectangles as possible into the larger one with the objective of minimizing the height, in polynomial time. Here, we will demonstrate how this can accomplished by using a the *Next-Fit Decreasing Height (NFDH)* approximation algorithm.

### Input and Output Specification

* Input

The program first receives, in one line, two positive integers and , where is the width of the texture atlas that the images will be packed into and is the number of images to be packed. The program then receives lines of input. Each line specifies an image and contains two positive integers and , the pixel height and width of the image, respectively.

* Output

The program outputs lines, specifying the positions of the images in the texture atlas. Each line contains two non-negative integers and , the and position of the bottom-left pixel of the image in the texture atlas, respectively. The positions are outputted in the same order as the input sizes are input; that is, the position in output corresponds to the image in input.

# Chapter 2: Algorithm Specification

* 1. Data Structures
* Min Heap

A min heap is used in the implementation of heap sort, and the resulting sequence is stored into an array.

* 1. Algorithm Specifications
* The Main Idea

The NFDH algorithm is an off-line algorithm, where all input is considered before an output is produced. Every inputted rectangle is sorted by their height in decreasing order, where the tallest gets dealt with first and is placed in the left of the texture atlas. A *next-fit* approach is used to pack the shapes, where a rectangle is packed in if and only if it does not exceed the width requirement. If the next rectangle to be packed goes over the width, it will be placed above the previous shape and justified left. This algorithm tends to pack the rectangles level-by-level, and earlier levels cannot be accessed.

* Sorting the Input- buildheap(), sort(), pop()

To sort the inputted rectangles in decreasing order by their height, we use heap sort. First, we build a min-heap using the address of each rectangle as the node’s key. We then pop the root, place it into the array node[] and heapify accordingly.

# Chapter 3: Testing Results

* 1. Test Cases
  2. Correctness Testing
  3. Performance Testing

# Chapter 4: Analysis and Comments

* 1. Time Complexity
  2. Space Complexity
  3. Approximation Scheme

# Conclusion

# Appendix: Source Code

# Declaration

*We hereby declare that all the work done in this project titled “Project Five: Texture Packing” is of our independent effort as a group.*

# Duty Assignments

**Programmer:**

**Tester:**

**Report Writer:**

# References