DSA lab exercises

9 November 2022

Problem 1

When a share of common stock of some company is sold, the ***capital gain*** (or, sometimes, loss) is the difference between the share's selling price and the price originally paid to buy it. This rule is easy to understand for a single share, but if we sell multiple shares of stock bought over a long period of time, then we must identify the shares actually being sold. A standard accounting principle for identifying which shares of a stock were sold in such a case is to use a FIFO protocol—the shares sold are the ones that have been held the longest (indeed, this is the default method built into several personal finance software packages).

For example, suppose we buy 100 shares at $20 each on day 1, 20 shares at $24 on day 2, 200 shares at $36 on day 3, and then sell 150 shares on day 4 at $30 each. Then applying the FIFO protocol means that of the 150 shares sold, 100 were bought on day 1, 20 were bought on day 2, and 30 were bought on day 3. The capital gain in this case would therefore be 100*10 + 20*6 + 30*(−6), or $940.

Write a program that takes as input a sequence of transactions of the form "buy x share(s) at $y each" or "sell x share(s) at $y each," assuming that the transactions occur on consecutive days and the values x and y are integers. Given this input sequence, the output should be the total capital gain (or loss) for the entire sequence, using the FIFO protocol to identify shares.

Problem 2

A slicing floorplan is a decomposition of a rectangle with horizontal and vertical sides using horizontal and vertical cuts (see fig 1). A slicing floorplan can be represented by a binary tree, called a slicing tree, whose internal nodes represent the cuts, and whose external nodes represent the basic rectangles into which the floorplan is decomposed by the cuts (See fig. 2). The compaction problem is defined as follows. Assume that each basic rectangle of a slicing floorplan is assigned a minimum width *w* and a minimum height *h*. The compaction problem is to find the smallest possible height and width for each rectangle of the slicing floorplan that is compatible with the minimum dimensions

of the basic rectangles. Namely this problem requires the assignment of values H(v) and W(v) to each node v of the slicing tree such that:

Note : Height of rectangle means length of the rectangle.

```
W(v)  =  w                 if v is an external node whose basic
                           rectangle has minimum width w.

      =  max(W(y),W(z))    if v is an internal node associated with
                           a horizontal cut with left child y and
                           right child z.

      = W(y)+W(z)          if v is an internal node associated with
                           a vertical cut with the left child y and
                           right child z.


H(v)  =  h                 if v is an external node whose basic
                           rectangle has minimum height h.

      =  (H(y)+ H(z))    if v is an internal node associated with
                          a horizontal cut with left child y and
                          right child z.

      = max(H(y),H(z))          if v is an internal node associated with
                           a vertical cut with the left child y and
                           right child z.
```

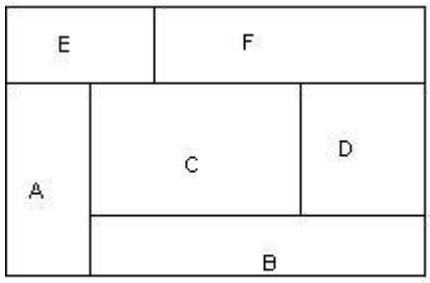

Fig. 1 Slicing Floorplan

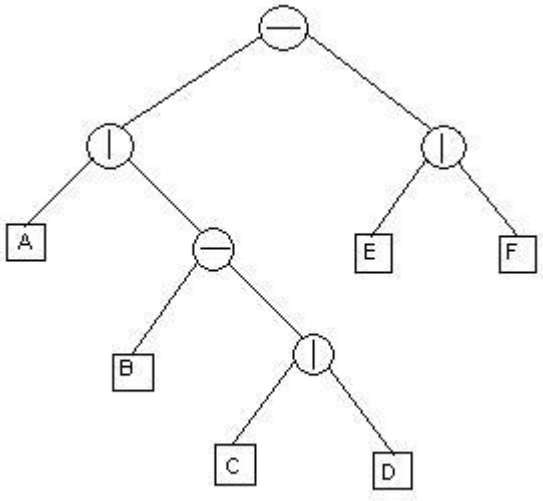

Fig. 2. Slicing Tree associated with the floorplan

Design a data structure in C++ for slicing floorplans that supports the operations.

1. Create a floorplan consisting of a single basic rectangle.
2. Decompose a basic rectangle by means of a horizontal cut.
3. Decompose a basic rectangle by means of a vertical cut.

4. Assign minimum height and width to a basic rectangle.
5. Draw the slicing tree associated with the floorplan.