Homework 17

Brian Knotten, Brett Schreiber, Brian Falkenstein

October 4, 2018

24

Define S_1 to be the set of boxes on the left side of the scale, and S_2 to be the boxes on the right side of the scale. The algorithm begins by sorting the boxes in order from least to most heavy. This can be done in $nlogn$ time using any fast sorting method (Merge, quick, etc). The decision tree can be defined as:

- The root of the tree represent the scale with all n boxes in S_1
- For a node v at depth i, take $v's$ left child to be moving $B_i + 1$ to S_2, and $v's$ right child to leave $B_i + 1$ in S_1.

The size of this tree will be 2^n. The depth will be n, as each depth i represents deciding over box B_i, and each node has a branching factor of 2^n.
The tree can be pruned with the following rules:

1. If a node v has a configuration of S_1 and S_2 such that $|S_1| < |S_2|$ (that is, the scale is tipped to the right), prune v.
2. If 2 nodes u and v at the same depth have the properties of $|S_1|_u = |S_1|_v$ and $|S_2|_u = |S_2|_v$, prune v.

The first pruning rule works because if we are deciding over boxes from least to most heavy, and never considering a box twice, a scale tipped to the right can never be reconciled, as we will only have heavier boxes to decide over in the future, and adding any boxes to S_2 will only make that side heavier. Thus a scale tipped to the right will always stay tipped to the right. If at any point we reach a state of equilibrium, where $|S_1| = |S_2|$, the algorithm terminates and returns true.

Consider the following 2-dimensional array A of size $[3^n, n]$, where $A[i, j] =$:

1. i is a ternary number where the kth bit of i represents B_k either
   (a) B_k is excluded from the equation.
   (b) B_k is on left hand side of the equation.
   (c) B_k is on the right hand side of the equation.

1. j represents the position of the operator.
2. The cell value is one of -1, 0, or 1, and it represents the left hand side is less than the right hand side, the sides are equal, and the left hand side is greater than the right hand side, respectively.

So, for example, with 5 total boxes, if $B_2 + B_4 > B_5$, then $A[01012, 2] = 1$, since B_1 is omitted from the equation, B_2 is on the left hand side, B_3 is omitted, B_4 is on the left hand side, B_5 is omitted from the equation, the operator is placed after the 2nd term, and the operator is the greater-than sign.

The dynamic algorithm makes $O(n^3)$ weighings on the scale as follows:

```
for i = 1 to n do:
    for j = 1 to n do:
        for k = 1 to (i - j) do:

            Place B[i] through B[i - k] on the left hand side of the scale.
            Place B[i - k + 1] through B[j] on the right hand side of the scale.

            A[current_assignment, k] =
                                       -1 if the scale tips left
                                        0 if the scale is level
                                        1 if the scale tips right

            where current_assignment is the encoding of the equation as a ternary number as described above
```

We can derive other equations algebraically without having to make any more measurements on the scale. For example, if it is known that $B_3 < B_4 + B_5$, and $B_1 + B_2 = B_3$, then substitution yields $B_1 + B_2 < B_4 + B_5$. We can continue to populate the 2-dimensional array and derive the comparisons that the scale can make without having to use the scale. The solution will be found at any of the cells whose ternary encoding does not contain a 0, that is, any comparison cell that utilizes all the boxes. If any of these cells is 0, then there exists an arrangement of all the boxes which balances the scale. Otherwise, there isn't.