6.891, Theory of Computing Machinery, Lecture 4, February 19, 1986. Lecturer: Charles Leiserson. Scribe: Bradley C. Kuszmaul.

Divide and Conquer Layout Strategies

Today we are going to discuss 'divide and conquer' layout strategies. To get started, we will consider the layout of a complete binary tree.

There are several ways to layout complete binary trees. The most obvious layout is not the best. Figure 1 shows a 'naive' layout of a complete binary tree. The layout is $O(n)$ wide, and $O(\log n)$ tall, giving an aera of $O(n \log n)$. A much better layout is the *H-tree* layout, which is shown in Figure 2. The H-tree layout has area $O(n)$, which is optimal since there are $O(n)$ vertices in the tree.

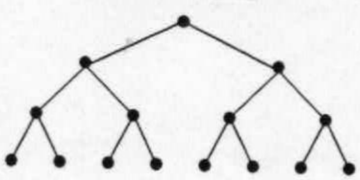

Figure 1: A naive $O(n \log n)$ binary tree layout.

One question that comes to mind is 'why is the H-tree layout more efficient?' There are several answers. The first answer is 'the area is smaller', but that does not tell us much. Two issues that seem to be more illustrative of what is going on are:

Aspect ratio: We note that the aspect ratio of the H-tree layout is more balanced. Asymptotically,

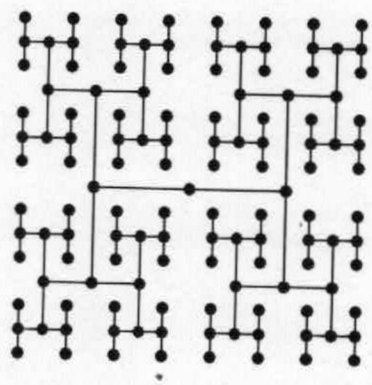

Figure 2: The H-tree layout with area $O(n)$.

the naive layout becomes very long compared to its length, while the H-tree stays square. We can look at what happens to the wire length.

Wire length: The wire lengths grow differently in the two layouts. In the naive layout, the wire lengths double every time we move one jump closer to the root of the tree. In the H-tree layout, the wire lengths double only every other jump.

It turns out that the way the wire length grows is very important. To understand this, we will look at several recurrence relations and see how the different growth rates affect the solutions to the relations.

With the naive layout, we have a recurrence relation as follows

$$A(n) = 2A(\lfloor n/2 \rfloor) + O(n) \tag{1}$$

which has a solution of $O(n \log n)$. This recurrence relation arises from the fact that there are two subtrees of size $n/2$, each of which has area $A(\lfloor n/2 \rfloor)$, plus we need $O(n)$ more wire to connect the top roots of the two subtrees with another node (since the roots of the subtrees are distance $O(n)$ apart).

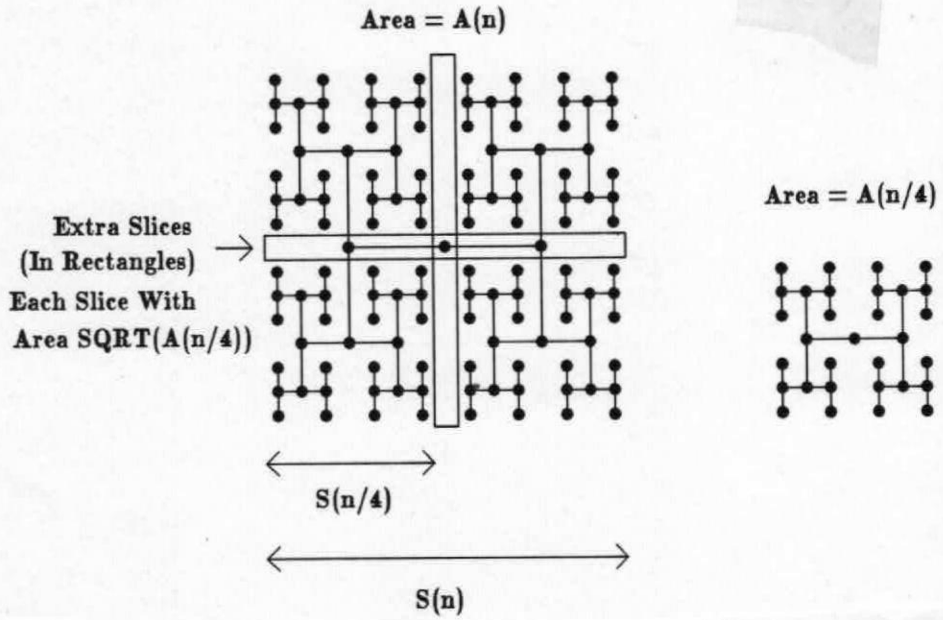

Figure 3: How to derive the recurence relation for the area of an H-tree.

The H-tree layout, on the other hand, has the following recurrence relation.

$$A(n) = 4A(\lfloor n/4) + 4\sqrt{A(\lfloor n/4\rfloor)} + 1 \tag{2}$$

This relation is derived as shown in Figure 3. Note that the H-tree in Figure 3 has been divided into four large pieces with a small slice between them. The first term in Equation 2 accounts for the area of the four large pieces, each of which contains $n/4$ of the nodes of the H-tree. The second term, $4\sqrt{A(\lfloor n/4\rfloor)}$, accounts for the four slices between the four small squares, and the extra constant value, 1, accounts for the very small square in the center of the layout.

Equation 2 looks very difficult to solve, but if we substitute $S(n) = \sqrt{A(n)}$ into Equation 2, we get

$$S^2(n) = 4S^2(\lfloor n/4\rfloor) + 4S(\lfloor n/4) + 1, \tag{3}$$

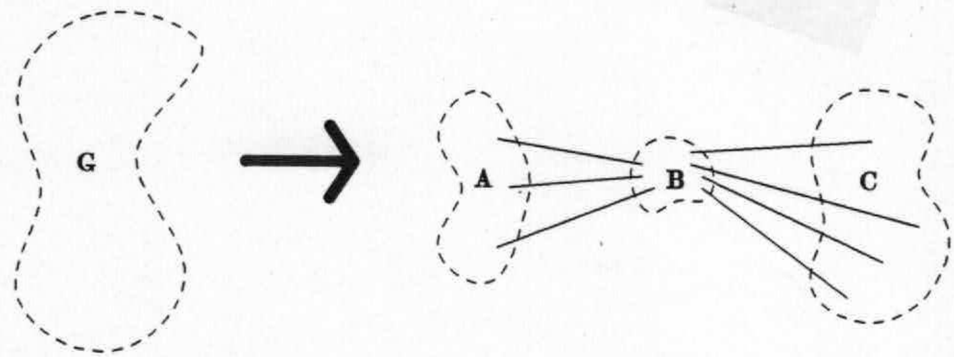

Figure 4: An example of a separator on G.

1. *No edge connects A to C,*
2. $|B| < cf(n)$, *and*
3. $|A| \leq |C| \leq (1-\alpha)n$.

Figure 4 shows an example of a graph partioned into a vertex separator. Note that $|B|$ is $O(f(n))$ and that the sizes of A and C are 'reasonably' equally balanced.

Definition 1 defines a *vertex separator*. A related question is to find edge separators, i.e. to find some set of edges which are the only connections between A and C, where A and C are reasonably equally balanced and the number of edges is $O(f(n))$. It is pretty clear that for edges of bounded degree, an $f(n)$ vertex separator gives an $f(n)$ edge separtor (just take the incident edges to B. There can not be too many of them), and the reverse is true (take the vertices which are incident to the edges forming the partition). Therefore, an $f(n)$ vertex separator is the same as an $f(n)$ edge separator for graphs of bounded degree. A star graph (see Figure 5) is an example where the two kinds of separators are very different. It takes $O(1)$ vertices to cut the graph into equal parts, but $O(n)$ edges.

Lipton and Tarjon showed that

- planar graphs have a $\sqrt{n}$ separator theorem, and

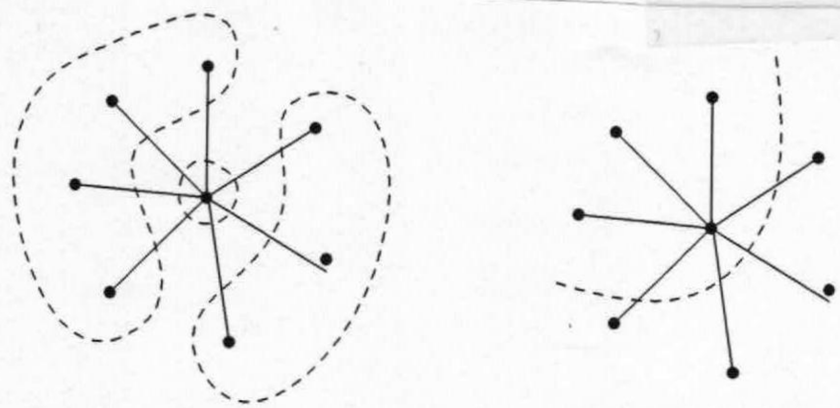

Figure 5: Edge separators are different from vertex separators for star graphs.

- trees have a 1 separator theorem. (Note that trees are not closed under the subgraph relation, but forests of trees are. In some sense, the business about the subgraph relation is a red herring.) (This can be proved with $\alpha = \frac{1}{3}$.)

The homework due on March 5, 1986 is to show that outerplanar graphs have a 1 separator theorem, and show that you can produce the separator in linear time.

You may want to refer to a text, such as Harary's book on graph theory.

Definition 2 *An* outerplanar *graph is a planar graph that you can embed into a plane such that all the vertices are on a single face (i.e. you can 'get to' all the vertices from the outside).*

A maximal outerplanar graph is shown in Figure 6. The definition of a maximal outerplanar graph is an outerplanar graph in which if one more edge is added, the graph would no longer be outerplanar. In general, maximal outerplanar graphs are triangulations of n-gons.

Note that trees are outerplanar graphs, since you can get to every vertex of the tree from the outside.

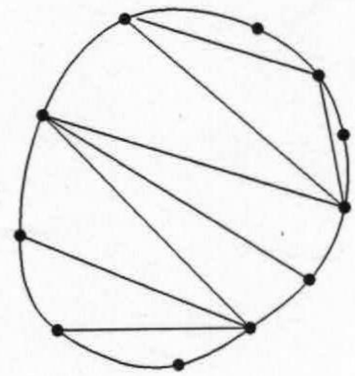

Figure 6: A maximal outerplanar graph.

Colinear Layouts

A natural 'easier' problem is the problem of colinear layouts.

Definition 3 *A layout is* colinear *if all the vertices of the graph lie on a single line.*

We have the following theorem about colinear layouts of binary trees.

Theorem 1 *Any binary tree can be laid out in $O(n \log n)$ area as a collinear layout.*

Proof: From Lipton and Tarjan we know that we can get a 1 separator theorem with $\frac{1}{4} \leq \alpha$. If we lay out the graph as shown in Figure 7 with a separating edge between two subpieces, and the two sub-pieces of the graph each have height less than $H((1-\alpha)n)$, then we need at least one more line of height to connect up the two sub-pieces, giving the following recurence relation:

$$H(n) \leq H(3/4n) + 1 \tag{6}$$

which has solution $O(\log n)$. (Here we are using an edge separator, but binary trees are of bounded degree, so we are ok.)

In general, a $f(n)$ separator theorem, with constants α and c, gives us height

$$H(n) \leq H((1-\alpha)n) + f(n) \tag{7}$$

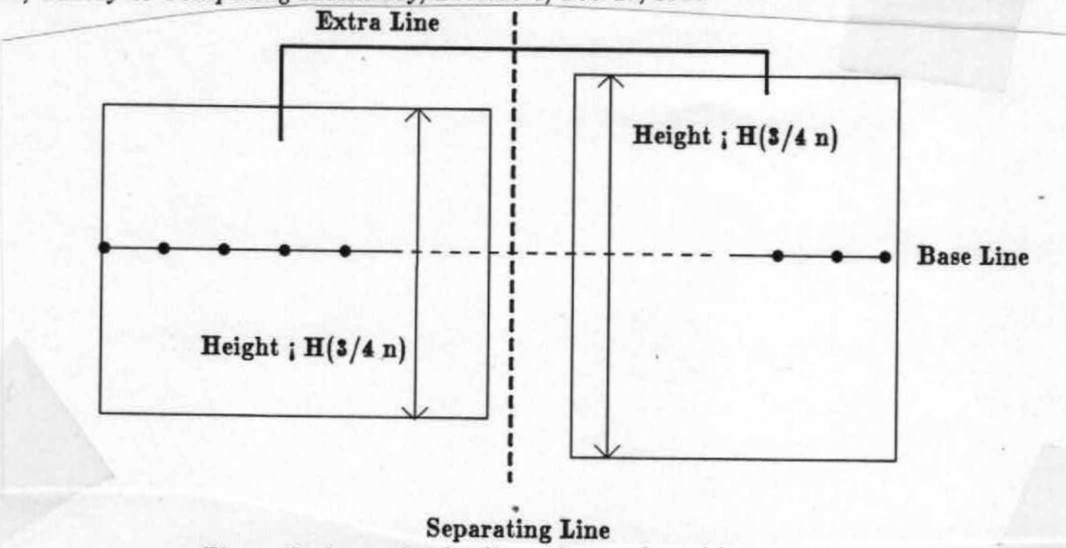

Figure 7: An optimal colinear layout for a binary tree.

which gives

$$H(n) = \begin{cases} O(n^\epsilon) & \text{if } f(n) = O(n^\epsilon)\text{, for some } \epsilon > 0\text{, then we get } n^\epsilon + (cn)^\epsilon + (c^2 n)^\epsilon + \cdots, \\ & \text{which is geomeetric and } c < 1 \text{ implies } O(n^\epsilon) \\ \Theta(\log^{k+1} n) & \text{if } f(n) = \Theta(\log^k n),\ k \geq 0\text{, since this forms an arithmetic series} \\ & \text{from one up to } \log^k n. \end{cases} \tag{8}$$

Note: One can ask what happens for things like $\log\log n$ and so forth, but we will stop at $\log^k n$.

Equation 7 automatically gives us a bound colinear layouts of planar graphs of $O(n^{3/2})$ area (since $n^{1/2}$ is the height and n is the width). This is a tight bound for planar graphs, (example for the bounding box area, a mesh must have $\sqrt{n}$ wires crossing any bisection).

Claim 1 *A 'naive' (read 'easy') bound for bounded degree graphs is n^2 area, since we can simply build δ large crossbar switches to implement the interconnections, where δ is the degree of the graph.*

We will show that this bound for trees is existentially tight. (I.e. there is a graph which requires this bound.) To show that the bound is universally tight is harder. We will show that

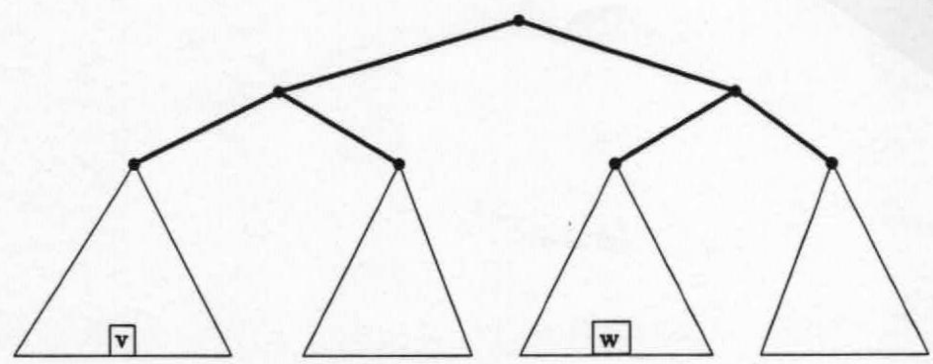

Figure 8: A complete binary tree cut into four subtrees.

there is a tree which takes $\Omega(n \log n)$ area for a collinear layout. The tree is the complete binary tree. We will prove the bounding box area meets this requirement, by induction on $n = 2^k - 1$ vertices.

Proof: The inductive hypothesis is that there is a perpendicular to the base line between the leftmost and rightmost vertices that cuts off at least $\lceil k/2 \rceil$ edges or vertices.

The base case involves checking for $n = 1$ and $n = 3$.

To do the inductive step, we cut the tree into four subtrees, as shown in Figure 8. Note that Figure 8 shows the tree, not the layout (since for example, there is no reason to believe *a priori* that the vertices of each of the four subtrees will not be mixed together). Let v be the leftmost vertex on the baseline (say that v is as shown in Figure 8). Let w be the rightmost vertex in a different subtree. The other two subtrees are wholly embedded between v and w. Choose one of the remaining subtrees. By inductive hypothesis, there is a perpendicular, l, that cuts at least $\lceil (k-2)/2 \rceil$ edges and vertices. There is a path from v to w that cuts l, so we must add one for that line. This gives us the relation $H(2^k) = \Omega(H(2^{k-2}) + 1)$, which proves the bound to be