COS 423, February 27, 1996
Handout Number 6

Some Remarks

Announcement Midterm exam will be given in class on March 26.

In this note, I will elaborate on some small details in the proof of the $O((m+n)\lg^* n)$ bound on the cost of executing a sequence σ of FINDs and UNIONs.

A Why σ' and σ have the same cost?

Intuitively this is obvious. But let us give a sketch of a formal proof. Let $\mathcal{F}_i$ denote the forest immediately after executing the ith instruction in σ. Let $\mathcal{F}_0$ be the forest consisting of n trees each consisting of a single element.

Let $\mathcal{F}'_0$ denote the forest obtained by executing all the Unions in σ' (but before any of the partial Finds). Each edge is of the form $(u, parent(u))$ coming from some Union instruction, and assigned a bit *on* or *off*. In $\mathcal{F}'_0$, every edge has its bit *off*. Let $\mathcal{F}''_0$ be the subforest of $\mathcal{F}'_0$ consisting only of edges with their bits *on*. Thus, $\mathcal{F}''_0$ is the disjoint union of n isolated elements, and thus identical to $\mathcal{F}_0$.

Consider the following transformation on $\mathcal{F}'_0$ induced by executing σ, for $i = 1, 2, \cdots$. Let $\mathcal{F}'_{i-1}$ be the forest before the ith instruction is carried out, and $\mathcal{F}''_{i-1}$ its subforest consisting of only *on* edges. Inductively, $\mathcal{F}''_{i-1}$ is identical to $\mathcal{F}_{i-1}$. If the ith instruction is $Union(u, v)$, then let $\mathcal{F}'_i$ be $\mathcal{F}'_{i-1}$ with the edge (u, v) turned *on*, and $\mathcal{F}''_i$ be its subforest consisting of only *on* edges. If the ith instruction is $Find(u)$, let v be the root of the tree containing u in $\mathcal{F}_{i-1}$. Then there is in σ' a corresponding instruction $Find(u, v)$. Carry out this partial find on $\mathcal{F}'_{i-1}$ to obtain $\mathcal{F}'_i$. Let $\mathcal{F}''_i$ be the subforest consisting of only *on* edegs. Since $\mathcal{F}''_{i-1}$ is identical to $\mathcal{F}_{i-1}$, the partial find compresses the same path in $\mathcal{F}'_{i-1}$ as in $\mathcal{F}_{i-1}$. In either of the above cases, $\mathcal{F}''_i$ is identical to $\mathcal{F}_i$. Furthermore, if the instruction is a Find, then the Partial find has the same cost as the corresponding Find. This completes the inductive proof.

B Charging FINDs to the nodes

Consider the execution of FIND(u, v), with the path being $u = x_0, x_1, x_2, \cdots, x_\ell, v$ in the current forest. By definition, the cost of this FIND is $\ell+1$. We charge a unit to each x_i such that $0 \le i < \ell$ and that x_i and $parent(x_i)$ belong to the same "group". (Note that x_ℓ is not charged.) We charge the rest of the $\ell+1$ units of cost to the FIND(u,v) instruction itself; this is either equal to or one less than 1+number of x_i with x_i, $parent(x_i)$ being in different groups,

and hence it is no greater than $1 + \lceil \lg^* n \rceil$. Thus, the total cost of executing σ is

$$O(1) \cdot N_U + O(\lceil \lg^* n \rceil) \cdot N_F + O(\sum_x c_x) = O(m \lg^* n + \sum_x c_x),$$

where N_U, N_F are the number of unions and finds in σ, and c_x is the number of units element x is charged with during the entire execution.

A critical observation is that $c_x \leq F(k) - F(k-1) \leq F(k)$ where k is the group x belongs to. The reason is that each time x gets charged one unit in a FIND, parent(x) gets changed from some x' to a node y with $rank(y) > rank(x')$ (y is in fact an ancestor of x'). This means, after x gets charged $F(k) - F(k-1)$ units, parent(x) will be a node with rank greater than $F(k)$. From that point on, $parent(x)$ always will be an element in group $k+1$ or higher, and thus x no longer gets charged even if x is on some find path.

The above discussion leads to $\sum_x c(x) \leq n \lg^* n$ as demonstrated in class (and in the handout from [AHU]).