Second Level Algorithms

Prof. Palash Dey

Department of Computer Science and Engineering

Indian Institute of Technology, Kharagpur

Week – 03

Lecture 12

Welcome to the twelfth lecture of the second-level algorithms course. In the last lecture, we have seen that the amortized time complexity of insertion is O(1), the amortized time complexity of extract min is O(log n), and the amortized time complexity of decrease key is O(1). There, we have assumed that the maximum degree of any node in an n-node Fibonacci heap is O. So, we will prove that result today. Let us begin.

So, here is the theorem that we will prove today. So, we will break this proof into a couple of lemmas. Our first lemma is as follows, which bounds the degree of any child node of any Fibonacci heap. So, here is the lemma. Let x be any node in a Fibonacci heap. We will assume that n is the number of nodes. So, we will avoid writing 'n-node Fibonacci heap' every time; we will simply write 'Fibonacci heap,' with the underlying assumption that the number of nodes in the Fibonacci heap is n, ok? So, I take an arbitrary node in the Fibonacci heap and suppose $x.degree$—recall, in every node, we are storing the degree, the number of children the node has—this number be k. Let $y_1, y_2, \cdots, y_k$ be the children of x in the order this is very important in the order in which they are made. children of x ok. That means, y_1 is the first node among this k nodes which are made the children of x at first then followed by y_2 followed by y_3 and so on that is y_1 is the earliest child among the children of x it may be possible that there is some node which was made the child of x before y_1. the decrease k operation may have cut the node y_0. So, that may be possible. So, among the so, let us look at a snapshot of the Fibonacci heap let us pick an arbitrary node x and currently it has k children among those k children let us name them in the order they are made the child ofWW x. That means, y_1 is the earliest children earliest child followed by y_2 and so on and y_k is the latest child among the children of x. we claim that $y_1.degree$ is of course, greater than 0 degree cannot be negative and interestingly $y_i.degree$ is greater than equal to $i-2$ for all $i=2,3,\ldots,k$ ok. So, let us prove this. $y_1.degree$ is greater than equal to 0 there is nothing to prove because degree cannot be negative. So, let us write since degree cannot be negative y_1 is greater than equal to 0 this fine.

So, now, let us put the prove the second part for $i=2,3,\ldots,k$ $y_i.degree$ is at least $i-2$. So, for $i=2,3,\ldots,k$ observe that when y_i was made a child of x, $y_1,\cdots,y_{i-1}$ where already children of x that is when y_i was made a child of x, the degree of x was at least $i-1$. Why at least why not exact it may be possible that x has lost 1 child in that x has lost 1 child after y_i has was made the child of x. So, at that time the degree of x could have been i also, but all we know is that y_1 to y_{i-1} all those nodes must be a child of x. we make one node a child of another only in the consolidate operation in the extract mean procedure. And we consolidate only in the consolidate we merge two sub trees or two root nodes right make one root node a child of another only if the degree of both the root nodes are same. when y_i was made a child of x then $y_i.degree$ of y_i. was the same as the degree of x at that point which was at least i minus 1 ok. Since then y_i could have lost at most one child because if it whenever it loses the second child it will it would have been cut from x and made it part of root list. The fact that y_i is not cut from x. it is still a child that means, that y_i can lose at most 1 child from that point.

Since then, y can lose at most 1 child. Hence, the current degree of y, which is $y_i.degree$, is greater than or equal to $i-2$, which we need to prove. So, this proves the lemma. The second one is about Fibonacci numbers. So, let us recall what Fibonacci numbers are.

Fibonacci numbers are defined recursively. So, f_i is f_0 is 0, f_1 is 1, and from f_2 onwards, f_i is $f_{i-1}+f_{i-2}$. So, our next lemma is this. So, this lemma is regarding Fibonacci numbers.

So, let us prove this lemma. So, we will prove by induction on k. So, if k equals 0. So, the base case is k equals 0. This obviously holds because LHS is f_2, which is 1, and RHS is also 1. So, the induction hypothesis assumes for $k-1$.

That is f_{k+1} is this ok. Now, let us prove the inductive step. f_{k+2} that is LHS from the definition of Fibonacci number every Fibonacci number is the sum of previous two Fibonacci numbers. So, we use that. the second term I leave as it is and for the first term I use the induction hypothesis.

What we have to prove RHS ok. So, this proves the lemma. Now, we need another lemma involving Fibonacci numbers which shows that Fibonacci numbers grows exponentially. f_{k+2} is greater than equal to this. So, let us prove this Fibonacci this lemma involving Fibonacci numbers.

Again we will prove by induction on k. So, the base case K is 0 LHS is 1 which is equal to RHS. Inductive hypothesis So, we assume for $k-1$ that is this and now the inductive step. LHS is f_{k+2}.

Now, we use the previous lemma, which is this: f_{k+2} is this f_k. And now we use the induction hypothesis, the inductive hypothesis. So, this is greater than or equal So, this is 1; this should not be k, this should be ϕ^{i-2} , and this starts Actually, it is even better to use the definition of the Fibonacci number.

There, we will get it straight away: f_{k+2} is $f_{k+1}+f_k$. The Fibonacci number is the sum of the last two Fibonacci numbers. Now, let us use the inductive hypothesis. The first term is bounded by and the second term is bounded by this. Now, $\phi+1$ is the same as ϕ^2; that is easy directly from the value. So, I will give it as homework. So, this is the right-hand side. So, this proves the lemma. This shows that the Fibonacci number grows exponentially.

The value of ϕ is approximately 1.619. ϕ is approximately 1.619. So, now we prove the final result. The number of nodes in a subtree rooted at x. Is at least the $(k+2)$-th Fibonacci number, which is at least ϕ^k , where k is the degree of x.

Proof. So, we define S_i for natural numbers i, which include 0 also. To be the minimum number of nodes in a subtree Rooted at a node of degree i, OK. So, all I need to show is that S_i or S_k is greater than or equal to f_{k+2} . So, here is a node, and suppose its degree is k This is $y_1, y_2, \ldots, y_k$; these are in the order in which these children are made a child of the parent node. So, then S_k is greater than or equal to S_k, which is the number of nodes, the minimum number of nodes in this subtree. So, S_k is this node itself, 1, and the degree of y_1 could be 0 also, at least 0.

So, these two black mark nodes. So, 2 plus now what else? What are the number of nodes in the subtree rooted at i? What is the minimum number of nodes that is $i=2,\ldots,k$ $y_i.degree$ right. And we have shown that the degree of $y_i \geq i-2$. So, this is at least 2 plus

OK, and now we show by induction that $S_i \geq S_{i-2}$. So, we use this claim that $S_i \geq f_{i+2}$. So, let us use this claim and let us continue this proof. So, this is greater than or equal to $2+\sum_{i=2}^{k} f_i$, which is same as $1+\sum_{i=0}^{k} f_i$ because the 0th Fibonacci number is 0, first Fibonacci number is 1.

This is f_i. Now, this expression we know is f_{k+2}. This is what we need to show: that the minimum number of nodes in a subtree rooted at a node whose degree is k is at least f_{k+2}. Hence, the size Hence, the number of nodes in a subtree rooted at x of degree k must be at least f_{k+2}, which is at least f_{k+2}. So, this proves this lemma modulo this claim, which we will see in the next lecture.

So, let us stop here. Thank you.