Lecture 12, Oct 18, 2023

Disjoint Sets (Continued)

- Recall that we can optimize a disjoint set forest using weighted union (WU) and path compression (PC) techniques; what is the complexity of a sequence σ of $n-1$ Union operations and $m > n$ Find operations?
- Define the function 2^{*n}:
  - $2^{*0} = 1$
  - $2^{*n+1} = 2^{2^{*n}}$ for $n \geq 0$
  - This function grows as: $2^{*0} = 1, 2^{*1} = 2, 2^{*2} = 4, 2^{*3} = 16, 2^{*4} = 65536, 2^{*5} \approx 1 \times 10^{19\,729}, \dots$
- Define the inverse 2^{*n} as $\log^* n = \min\left\{ k : 2^{*k} \geq n \right\}$, i.e. the first value of k such that 2^{*k} is greater than n
  - For all intents and purposes this is basically constant
- We claim that with WU and PC, executing σ takes $O(m \log^* n)$ time (proof is left as an exercise to the reader)
- Is there some sequence of σ that takes at least $m \log^* n$ time? Can we execute σ in $O(m)$ time?
  - The real complexity is actually between the two – not quite linear, but growing slower than $m \log^* n$
  - The actual complexity is $\Theta(m\alpha(m, n))$ where $\alpha(m, n)$ is the inverse Ackermann function
  - The lower bound of $\Omega(m\alpha(m, n))$ applies for any disjoint set data structure, not just the forest implementation
  - This took 25 years to derive