CS4311
Design and Analysis of Algorithms

Tutorial: Analysis of Union-By-Rank with Path Compression

About this lecture

- Introduce an Ackermann-like function which grows even faster than Ackermann
  - Use it to represent "relative level" of difference between two numbers

- Analyze the amortized complexity of Union-By-Rank + Path Compression
  - Based on the above function

Review: Ackermann Function

- The Ackermann function is defined recursively as follows:

$$A_0(n) = n + 1$$

$$A_m(0) = A_{m-1}(1)$$

$$A_m(n) = \underbrace{A_{m-1}(A_{m-1}(\ldots A_{m-1}(A_{m-1}}_{n+1 \text{ iterations of } A_{m-1}}(1))\ldots)$$

A Similar Function

- We now define a similar function:

$$B_0(n) = n + 1$$

$$B_m(n) = \underbrace{B_{m-1}(B_{m-1}(\ldots B_{m-1}(B_{m-1}}_{n+1 \text{ iterations of } B_{m-1}}(n))\ldots)$$

observe the difference

- By induction, we can show that for all $n \geq 1$ and all m,

$$B_m(n) \geq A_m(n)$$

$\beta(n)$: Inverse of B

- The inverse of Ackermann is defined as:

$$\alpha(n) = \alpha(n,n) = \min \{ k \mid A_k(1) \geq \log n \}$$

- We can define inverse of B similarly :

$$\beta(n) = \min \{ k \mid B_k(1) \geq n \}$$

- Like $\alpha(n)$, $\beta(n)$ is at most 4 in practice
- In fact: $\beta(n) = O(\alpha(n))$

Relative Level (1)

- Let X, Y be positive integers, with $X > Y$
- We can use the previous function B to measure how much X is larger than Y :
  - Precisely, we find the largest ℓ with

$$B_\ell(Y) \leq X$$

  - We call this ℓ the level of X w.r.t. Y
- Intuitively, think of X and Y as the money possessed by two very rich persons ; the level measures how far away one is from the other ...

Relative Level (2)

- Suppose ℓ = the level of X w.r.t. Y
- Based on ℓ, we can further define the degree of difference between X and Y
  - Precisely, we find the largest d with

$$\underbrace{B_\ell(B_\ell(\ldots B_\ell(B_\ell}_{d \text{ iterations of } B_\ell}(Y))\ldots) \le X$$

Lemma: $1 \le d \le Y$ (why??)

Analysis of Amortized Cost

- We are ready to analyze the amortized cost of Union/Find operations when we apply Union-By-Rank + Path Compression

- Recall that each node stores a rank value
  - Our potential function is based on the relative level between the rank of a node with the rank of its parent

- We first give some properties of rank

Properties of Rank

Suppose the input set has n elements

Property 1: Once a node has got a parent, its rank will never change

Property 2: Rank of a node $\leq$ n - 1

Property 3: For any node with a parent, rank of a node $<$ rank of its parent

Max Level

- Let rank(x) = rank of a node x
- Let u = node with rank at least 1,

  p = parent of u
- Let M = relative level between

  rank(u) and rank(p)

➔ $B_M(1) \le B_M(\text{rank}(u)) \le \text{rank}(p) \le n\text{-}1$

- On the other hand, $B_{\beta(n)}(1) \ge n$

➔ $M < \beta(n)$

Potential Function

- For our analysis, we use the following potential function :
  - For a root or rank-0 node u,

    $$\Phi(u) = \beta(n)\ \mathrm{rank}(u)$$

  - For other node v (rank at least 1),

    $$\Phi(v) = (\beta(n) - \ell_v)\ \mathrm{rank}(v) - d_v$$

  where ℓ_v = level of v w.r.t. its parent

  d_v = degree of v w.r.t. its parent

Potential Function

- Potential of forest F :

  $\Phi(F)$ = sum of potentials of all nodes

Lemma: $\Phi(F)$ is at least 0

Proof: For any non-leaf node v,

$\beta(n) - \ell_v$ is at least 1, and $d_v \leq \text{rank}(v)$

Amortized Cost for Find

Consider performing Find(x)

- Which nodes will change their potential ?

  Ans. x and all its ancestors (say, k of them)

- What will be the change their potential ?

  Ans. Most (except at most $\beta(n)$ of them) drops by at least 1 [See proof in next slides]

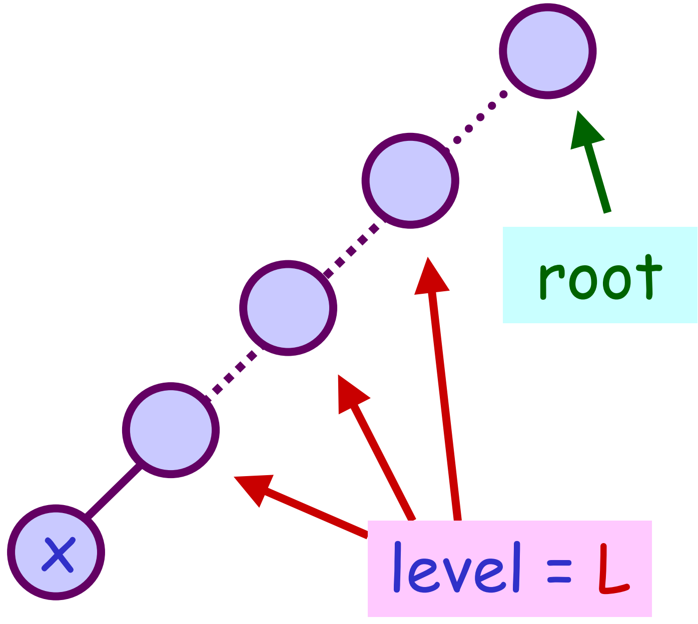
➔ amortized cost of Find

$$\leq O(k+1) - (k - \beta(n)) = O(\beta(n))$$

Proof

- Consider ancestors of x with the same level L

Lemma: Apart from the one closest to root, potential of each ancestor drops by 1 after Union



Proof: Let v be one of such ancestors.
There are two cases ...

Case 1 : ℓ_v increases

Let ℓ, d = level and degree before Union

ℓ^*, d^* = level and degree after Union

As $(\beta(n) - \ell^*)\,\text{rank}(v) - d^*$ ← $\Phi(v)$ after Union

$< (\beta(n) - \ell^*)\,\text{rank}(v)$ [as $d^* \geq 1$]

$\leq (\beta(n) - \ell)\,\text{rank}(v) - \text{rank}(v)$ [as ℓ_v increases]

$\leq (\beta(n) - \ell)\,\text{rank}(v) - d$ [as $\text{rank}(v) \geq d$]

$\Phi(v)$ before Union

➔ $\Phi(v)$ drops by at least 1

Case 2 : ℓ_v unchanged

Let d = degree before Union

- Before Union, some ancestor u of v has same level L as v

$\Rightarrow$ rank(root) $\geq$ rank(parent(u))

$\geq B_L$(rank(u)) $\geq B_L$(rank(parent(v)))

$\geq B_L(\underbrace{B_L \ldots (B_L}_{d \text{ of them}}(\text{rank}(v)) \ldots)$

$\Rightarrow$ d increases by at least 1 after Union

$\Rightarrow$ Φ(v) drops by at least 1

Key Result for Find

Corollary: Consider the ancestors of x .
When Find(x) is performed, except at most $\beta(n)$ of them, the potential of all other ancestors drops by at least 1

Proof: All by one node from each level drops the potential by at least 1

→ corollary follows since there are at most $\beta(n)$ distinct levels

Amortized Cost for Union

Consider performing Union(x,y) :

- Let r_x = root of tree containing x
- Let r_y = root of tree containing y
- Suppose that during Union(x,y),
  r_x is linked to r_y (making r_y the new root)

- Which nodes will change their potential ?

  Ans. r_x, r_y, children of r_y

Amortized Cost for Union

- Which nodes will increase their potential ?

$\Phi(r_x)$ after Union : always decreases

$\Phi(r_y)$ after Union : may increase if its rank increases ;

cannot decrease

Φ(child v of r_y) after Union : may decrease ;

cannot increase ;

➔ Only $\Phi(r_y)$ may increase

Amortized Cost for Union

- Precisely,

  $\Phi(r_y)$ may increase by at most $\beta(n)$

- Since Union can be performed by first Find(x), Find(y), and then link the roots,

  ➔ amortized cost of Union(x,y)

  $\leq$ 2 $\times$ amortized cost of Find +

  1 + increase in potential = $O(\beta(n))$