Упражнения

21.3.1

Выполните упр. 21.2.2 для леса непересекающихся множеств с использованием объединения по рангу и сжатия пути.

21.3.2

Напишите нерекурсивную версию процедуры FIND-SET со сжатием пути.

21.3.3

Приведите последовательность из m операций MAKE-SET, UNION и FIND- SET, n из которых — операции MAKE-SET, время выполнения которой составляет $\Omega(m \lg n)$ при использовании только объединения по рангу.

21.3.4

Предположим, что мы хотим добавить операцию PRINT-SET(x), которая для заданного узла x выводит все члены множества x в произвольном порядке. Покажите, как можно добавить к каждому узлу леса непересекающихся множеств по одному атрибуту, так чтобы время работы процедуры PRINT-SET(x) линейно зависело от количества членов множества x и при этом асимптотические времена работы других операций остались неизменными. Считайте, что каждый член множества выводится за время $O(1)$.

21.3.5 ★

Покажите, что время выполнения произвольной последовательности m операций MAKE-SET, FIND-SET и LINK, в которой все операции LINK выполняются до первой операции FIND-SET, равно $O(m)$, если используются как объединение по рангу, так и сжатие пути. Что будет, если воспользоваться только сжатием пути?

★ 21.4. Анализ объединения по рангу со сжатием пути

Как отмечалось в разделе 21.3, время работы m операций над непересекающимися множествами из n элементов при использовании объединения по рангу и сжатия пути равно $O(m\,\alpha(n))$. В данном разделе мы рассмотрим функцию α и выясним, насколько медленно она растет. Затем мы докажем приведенное выше время работы с использованием метода потенциала из амортизационного анализа.

Очень быстро растущая функция и очень медленно растущая функция, обратная к ней

Определим для целых $k \geq 0$ и $j \geq 1$ функцию $A_k(j)$ как

$$A_k(j) = \begin{cases} j+1 , & \text{если } k = 0 , \\ A_{k-1}^{(j+1)}(j) , & \text{если } k \geq 1 , \end{cases}$$

где выражение $A_{k-1}^{(j+1)}(j)$ использует функционально-итеративные обозначения из раздела 3.2. В частности, $A_{k-1}^{(0)}(j) = j$, а $A_{k-1}^{(i)}(j) = A_{k-1}(A_{k-1}^{(i-1)}(j))$ для $i \geq 1$. Параметр k будем называть ***уровнем*** функции A.

Функция $A_k(j)$ — строго возрастающая по j и k. Для того чтобы увидеть, насколько быстро растет данная функция, начнем с записи функций $A_1(j)$ и $A_2(j)$ в явном виде.

Лемма 21.2

Для любого целого $j \geq 1$ мы имеем $A_1(j) = 2j + 1$.

Доказательство. Воспользуемся индукцией по i, чтобы показать, что $A_0^{(i)}(j) = j+i$. Для базы индукции мы имеем $A_0^{(0)}(j) = j = j+0$. Для выполнения шага индукции предположим, что $A_0^{(i-1)}(j) = j+(i-1)$. Тогда $A_0^{(i)}(j) = A_0(A_0^{(i-1)}(j)) = (j+(i-1))+1 = j+i$. Наконец заметим, что $A_1(j) = A_0^{(j+1)}(j) = j+(j+1) = 2j+1$. ■

Лемма 21.3

Для любого целого $j \geq 1$ мы имеем $A_2(j) = 2^{j+1}(j+1) - 1$.

Доказательство. Воспользуемся индукцией по i, чтобы показать, что $A_1^{(i)}(j) = 2^i(j+1) - 1$. Для базы индукции мы имеем $A_1^{(0)}(j) = j = 2^0(j+1) - 1$. Для выполнения шага индукции предположим, что $A_1^{(i-1)}(j) = 2^{i-1}(j+1) - 1$. Тогда $A_1^{(i)}(j) = A_1(A_1^{(i-1)}(j)) = A_1(2^{i-1}(j+1)-1) = 2\cdot(2^{i-1}(j+1)-1)+1 = 2^i(j+1)-2+1 = 2^i(j+1)-1$. Наконец заметим, что $A_2(j) = A_1^{(j+1)}(j) = 2^{j+1}(j+1)-1$. ■

Теперь посмотрим, насколько быстро растет функция $A_k(j)$, просто вычисляя $A_k(1)$ для уровней $k = 0,1,2,3,4$. Из определения $A_0(k)$ и рассмотренных выше лемм мы имеем $A_0(1) = 1+1 = 2$, $A_1(1) = 2\cdot 1+1 = 3$ и $A_2(1) = 2^{1+1}\cdot(1+1) - 1 = 7$. Кроме того,

$$\begin{aligned}
A_3(1) &= A_2^{(2)}(1) \\
&= A_2(A_2(1)) \\
&= A_2(7) \\
&= 2^8 \cdot 8 - 1 \\
&= 2^{11} - 1 \\
&= 2047
\end{aligned}$$

и

$$\begin{aligned}
A_4(1) &= A_3^{(2)}(1) \\
&= A_3(A_3(1)) \\
&= A_3(2047) \\
&= A_2^{(2048)}(2047) \\
&\gg A_2(2047) \\
&= 2^{2048} \cdot 2048 - 1 \\
&> 2^{2048} \\
&= (2^4)^{512} \\
&= 16^{512} \\
&\gg 10^{80} ,
\end{aligned}$$

что представляет собой оценочное количество атомов в наблюдаемой Вселенной. (Символ "$\gg$" означает отношение "гораздо больше, чем".)

Определим обратную к $A_k(n)$ функцию для $n \geq 0$ следующим образом:

$$\alpha(n) = \min\{k : A_k(1) \geq n\} .$$

Другими словами, $\alpha(n)$ — наименьший уровень k, для которого $A_k(1)$ не меньше n. Исходя из найденных выше значений $A_k(1)$ мы видим, что

$$\alpha(n) = \begin{cases} 0 & \text{для } 0 \leq n \leq 2 , \\ 1 & \text{для } n = 3 , \\ 2 & \text{для } 4 \leq n \leq 7 , \\ 3 & \text{для } 8 \leq n \leq 2047 , \\ 4 & \text{для } 2048 \leq n \leq A_4(1) . \end{cases}$$

Только для невероятно больших, "астрономических" значений n (бóльших $A_4(1)$) значение функции $\alpha(n)$ превышает 4, так что для всех практических применений $\alpha(n) \leq 4$.

Свойства рангов

В оставшейся части этого раздела мы докажем, что при использовании объединения по рангам и сжатия пути граница времени работы операций над непересекающимися множествами равна $O(m\,\alpha(n))$. Для этого нам потребуются некоторые простые свойства рангов.

Лемма 21.4

Для всех узлов x мы имеем $x.rank \leq x.p.rank$, причем при $x \neq x.p$ выполняется строгое неравенство. Значение $x.rank$ изначально равно 0 и со временем увеличивается, пока не будет достигнуто $x \neq x.p$; после этого величина $x.rank$ не изменяется. Значение $x.p.rank$ монотонно растет со временем.

Доказательство. Доказательство выполняется простой индукцией по количеству операций с использованием реализаций процедур MAKE-SET, UNION и FIND-SET в разделе 21.3. Подробности доказательства оставлены в качестве упр. 21.4.1. ■

Следствие 21.5

При перемещении вдоль простого пути от произвольного узла к корню ранг строго возрастает. ■

Лемма 21.6

Ранг любого узла не превышает $n - 1$.

Доказательство. Начальный ранг каждого узла — 0, и он возрастает только при выполнении операции LINK. Поскольку выполняется не более $n - 1$ операции UNION, операций LINK также выполняется не более чем $n - 1$. Поскольку каждая операция LINK либо оставляет все ранги неизменными, либо увеличивает ранг некоторого узла на 1, ни один ранг не может превышать $n - 1$. ■

Лемма 21.6 дает слабую оценку границы рангов. В действительности ранг любого узла не превосходит $\lfloor \lg n \rfloor$ (см. упр. 21.4.2). Однако для наших целей достаточно границы, определяемой леммой 21.6.

Доказательство границы времени работы

Для доказательства границы $O(m\alpha(n))$ мы воспользуемся методом потенциала из амортизационного анализа (см. раздел 17.3). В ходе амортизационного анализа удобно считать, что мы выполняем операцию LINK, а не UNION. Иначе говоря, поскольку параметрами процедуры LINK являются указатели на два корня, мы считаем, что соответствующие операции FIND-SET выполняются отдельно. Приведенная далее лемма показывает, что, даже если учесть дополнительные операции FIND-SET, инициированные вызовами UNION, асимптотическое время работы останется неизменным.

Лемма 21.7

Предположим, что мы преобразуем последовательность S' из m' операций MAKE-SET, UNION и FIND-SET в последовательность S из m операций MAKE-SET, LINK и FIND-SET путем преобразования каждой операции UNION в две операции FIND-SET, за которыми следует операция LINK. Тогда, если последовательность S выполняется за время $O(m\,\alpha(n))$, то последовательность S' выполняется за время $O(m'\,\alpha(n))$.

Доказательство. Поскольку каждая операция UNION в последовательности S' преобразуется в три операции в S, выполняется соотношение $m' \leq m \leq 3m'$.

Так как $m = O(m')$, из границы времени работы $O(m\,\alpha(n))$ преобразованной последовательности S следует граница $O(m'\,\alpha(n))$ времени работы исходной последовательности S'. ■

В оставшейся части этого раздела мы полагаем, что исходная последовательность из m' операций MAKE-SET, UNION и FIND-SET преобразована в последовательность из m операций MAKE-SET, LINK и FIND-SET. Теперь докажем, что время выполнения полученной последовательности равно $O(m\,\alpha(n))$, и обратимся к лемме 21.7 для доказательства времени работы $O(m'\,\alpha(n))$ исходной последовательности из m' операций.

Функция потенциала

Используемая нами функция потенциала каждому узлу x в лесу непересекающихся множеств после выполнения q операций присваивает потенциал $\phi_q(x)$. Для получения потенциала всего леса мы суммируем потенциалы его узлов: $\Phi_q = \sum_x \phi_q(x)$, где Φ_q обозначает потенциал всего леса после выполнения q операций. До выполнения первой операции лес пуст, и мы полагаем, что $\Phi_0 = 0$. Потенциал Φ_q никогда не может стать отрицательным.

Значение $\phi_q(x)$ зависит от того, является ли x корнем дерева после q-й операции. Если это так или если $x.rank = 0$, то $\phi_q(x) = \alpha(n) \cdot x.rank$.

Теперь предположим, что после q-й операции x не является корнем и что $x.rank \geq 1$. Необходимо определить две вспомогательные функции от x до определения $\phi_q(x)$. Сначала определим

$$\text{level}(x) = \max\{k : x.p.rank \geq A_k(x.rank)\} ,$$

т.е. $\text{level}(x)$ — наибольший уровень k, для которого A_k, примененное к рангу x, не превышает ранг родителя x.

Мы утверждаем, что

$$0 \leq \text{level}(x) < \alpha(n) . \tag{21.1}$$

Это можно подтвердить следующим образом. Мы имеем

$$\begin{aligned} x.p.rank &\geq x.rank + 1 && \text{(согласно лемме 21.4)} \\ &= A_0(x.rank) && \text{(согласно определению } A_0(j)) , \end{aligned}$$

откуда вытекает, что $\text{level}(x) \geq 0$, и мы имеем

$$\begin{aligned} A_{\alpha(n)}(x.rank) &\geq A_{\alpha(n)}(1) && \text{(поскольку } A_k(j) \text{ строго возрастающая)} \\ &\geq n && \text{(согласно определению } \alpha(n)) \\ &> x.p.rank && \text{(согласно лемме 21.6)} , \end{aligned}$$

откуда вытекает, что $\text{level}(x) < \alpha(n)$. Заметим, что, поскольку $x.p.rank$ монотонно растет со временем, то же происходит и с $\text{level}(x)$.

Вторая вспомогательная функция применима при $x.rank \geq 1$:

$$\text{iter}(x) = \max\left\{i : x.p.rank \geq A^{(i)}_{\text{level}(x)}(x.rank)\right\} ,$$

т.е. функция $\text{iter}(x)$ представляет собой наибольшее количество итеративного применения функции $A_{\text{level}(x)}$ к исходному рангу x, до того как мы получим значение, превышающее ранг родителя x.

Мы утверждаем, что при $x.rank \geq 1$

$$1 \leq \text{iter}(x) \leq x.rank , \tag{21.2}$$

в чем можно убедиться следующим образом. Мы имеем

$$\begin{aligned} x.p.rank &\geq A_{\text{level}(x)}(x.rank) && \text{(согласно определению level}(x)) \\ &= A^{(1)}_{\text{level}(x)}(x.rank) , \end{aligned}$$

где последнее равенство следует из определения функциональной итерации. Отсюда вытекает, что $\text{iter}(x) \geq 1$, и мы имеем

$$\begin{aligned} A^{(x.rank+1)}_{\text{level}(x)}(x.rank) &= A_{\text{level}(x)+1}(x.rank) && \text{(по определению } A_k(j)) \\ &> x.p.rank && \text{(по определению level}(x)) , \end{aligned}$$

откуда следует, что $\text{iter}(x) \leq x.rank$. Заметим, что, поскольку $x.p.rank$ монотонно возрастает со временем, для того, чтобы значение $\text{iter}(x)$ уменьшалось, значение $\text{level}(x)$ должно возрастать. Пока значение $\text{level}(x)$ остается неизменным, значение $\text{iter}(x)$ должно либо увеличиваться, либо вообще не изменяться.

Имея описанные вспомогательные функции, мы можем определить потенциал узла x после q операций:

$$\phi_q(x) = \begin{cases} \alpha(n) \cdot x.rank , & \text{если } x \text{ является корнем} \\ & \text{или } x.rank = 0 , \\ (\alpha(n) - \text{level}(x)) \cdot x.rank - \text{iter}(x) , & \text{если } x - \text{не корень} \\ & \text{и } x.rank \geq 1 . \end{cases}$$

Далее рассмотрим некоторые полезные свойства потенциалов узлов.

Лемма 21.8

Для каждого узла x и для всех количеств операций q имеем

$$0 \leq \phi_q(x) \leq \alpha(n) \cdot x.rank .$$

Доказательство. Если x является корнем или $x.rank = 0$, то $\phi_q(x) = \alpha(n) \cdot x.rank$ по определению. Предположим теперь, что x не является корнем и что $x.rank \geq 1$. Мы получим нижнюю границу $\phi_q(x)$ путем максимизации $\text{level}(x)$ и $\text{iter}(x)$. Согласно границе (21.1) $\text{level}(x) \leq \alpha(n) - 1$, а согласно границе (21.2) $\text{iter}(x) \leq x.rank$. Таким образом,

$$\begin{aligned}\phi_q(x) &= (\alpha(n) - \text{level}(x)) \cdot x.rank - \text{iter}(x) \\ &\geq (\alpha(n) - (\alpha(n) - 1)) \cdot x.rank - x.rank \\ &= x.rank - x.rank \\ &= 0\,.\end{aligned}$$

Аналогично мы получаем верхнюю границу $\phi_q(x)$ минимизацией $\text{level}(x)$ и $\text{iter}(x)$. В соответствии с границей (21.1) $\text{level}(x) \geq 0$, а в соответствии с границей (21.2) $\text{iter}(x) \geq 1$. Таким образом,

$$\begin{aligned}\phi_q(x) &\leq (\alpha(n) - 0) \cdot x.rank - 1 \\ &= \alpha(n) \cdot x.rank - 1 \\ &< \alpha(n) \cdot x.rank\,.\end{aligned}$$ ∎

Следствие 21.9

Если узел x не является корнем и $x.rank > 0$, то $\phi_q(x) < \alpha(n) \cdot x.rank$. ∎

Изменения потенциала и амортизированная стоимость операций

Теперь мы готовы к рассмотрению вопроса о влиянии операций над непересекающимися множествами на потенциалы узлов. Зная, как изменяется потенциал при той или иной операции, мы можем определить амортизированную стоимость каждой операции.

Лемма 21.10

Пусть x — узел, не являющийся корнем, и предположим, что q-я операция — либо LINK, либо FIND-SET. Тогда после выполнения q-й операции $\phi_q(x) \leq \phi_{q-1}(x)$. Более того, если $x.rank \geq 1$ и из-за выполнения q-й операции происходит изменение либо $\text{level}(x)$, либо $\text{iter}(x)$, то $\phi_q(x) \leq \phi_{q-1}(x) - 1$. То есть потенциал x не может возрастать, и если он имеет положительное значение и либо $\text{level}(x)$, либо $\text{iter}(x)$ изменяется, то потенциал x уменьшается как минимум на 1.

Доказательство. Поскольку x не является корнем, q-я операция не изменяет $x.rank$, и так как n не изменяется после первых n операций MAKE-SET, $\alpha(n)$ остается неизменной величиной. Следовательно, эти компоненты формулы функции потенциала x после выполнения q-й операции не изменяются. Если $x.rank = 0$, то $\phi_q(x) = \phi_{q-1}(x) = 0$. Теперь предположим, что $x.rank \geq 1$.

Вспомним, что значение функции $\text{level}(x)$ монотонно растет со временем. Если q-я операция оставляет значение $\text{level}(x)$ неизменным, то значение $\text{iter}(x)$ либо возрастает, либо остается неизменным. Если и $\text{level}(x)$, и $\text{iter}(x)$ не изменяются, то $\phi_q(x) = \phi_{q-1}(x)$. Если же значение $\text{level}(x)$ не изменяется, а $\text{iter}(x)$ возрастает, то последнее значение увеличивается как минимум на 1, так что $\phi_q(x) \leq \phi_{q-1}(x) - 1$.

Наконец, если q-я операция увеличивает $\text{level}(x)$, то это увеличение составляет как минимум 1, так что значение члена $(\alpha(n) - \text{level}(x)) \cdot x.rank$ уменьшается как минимум на величину $x.rank$. Так как значение $\text{level}(x)$ возрастает, значение $\text{iter}(x)$ может уменьшаться, но в соответствии с границей (21.2) это уменьшение не может превышать $x.rank - 1$. Таким образом, увеличение потенциала, вызванное изменением значения $\text{iter}(x)$, меньше, чем уменьшение потенциала из-за изменения $\text{level}(x)$, так что мы можем заключить, что $\phi_q(x) \le \phi_{q-1}(x) - 1$. ■

Наши последние три леммы показывают, что амортизированная стоимость каждой из операций MAKE-SET, LINK и FIND-SET составляет $O(\alpha(n))$. Вспомним, что согласно (17.2) амортизированная стоимость каждой операции равна ее фактической стоимости плюс увеличение потенциала, вызванное ее выполнением.

Лемма 21.11

Амортизированная стоимость каждой операции MAKE-SET равна $O(1)$.

Доказательство. Предположим, что q-й операцией является MAKE-SET(x). Эта операция создает узел x с рангом 0, так что $\phi_q(x) = 0$. Никакие иные ранги и потенциалы не изменяются, так что $\Phi_q = \Phi_{q-1}$. То, что фактическая стоимость операции MAKE-SET равна $O(1)$, завершает доказательство. ■

Лемма 21.12

Амортизированная стоимость каждой операции LINK равна $O(\alpha(n))$.

Доказательство. Предположим, что q-й операцией является LINK(x, y). Фактическая стоимость операции LINK равна $O(1)$. Без потери общности предположим, что LINK делает y родительским узлом x.

Для определения изменения потенциала из-за выполнения операции LINK заметим, что множество узлов, потенциалы которых могут измениться, ограничено узлами x, y и дочерними узлами y непосредственно перед операцией. Мы покажем, что единственным узлом, потенциал которого может увеличиться в результате выполнения операции LINK, является узел y, и это увеличение не превышает $\alpha(n)$.

- Согласно лемме 21.10 потенциал любого узла, являющегося дочерним узлом y перед выполнением операции LINK, не может увеличиться в результате выполнения этой операции.
- По определению $\phi_q(x)$ мы видим, что, поскольку узел x непосредственно перед q-й операцией был корнем, $\phi_{q-1}(x) = \alpha(n) \cdot x.rank$. Если $x.rank = 0$, то $\phi_q(x) = \phi_{q-1}(x) = 0$. В противном случае

$$\begin{aligned} \phi_q(x) &< \alpha(n) \cdot x.rank && \text{(согласно следствию 21.9)} \\ &= \phi_{q-1}(x) \,, \end{aligned}$$

  так что потенциал x уменьшается.

- Поскольку y непосредственно перед выполнением операции LINK был корнем, $\phi_{q-1}(y) = \alpha(n) \cdot y.rank$. Операция LINK оставляет y корнем и либо оставляет ранг y неизменным, либо увеличивает его на 1. Следовательно, либо $\phi_q(y) = \phi_{q-1}(y)$, либо $\phi_q(y) = \phi_{q-1}(y) + \alpha(n)$.

Таким образом, увеличение потенциала вследствие операции LINK не превышает $\alpha(n)$, и амортизированная стоимость операции LINK равна $O(1) + \alpha(n) = O(\alpha(n))$. ■

Лемма 21.13

Амортизированная стоимость каждой операции FIND-SET равна $O(\alpha(n))$.

Доказательство. Предположим, что q-й операцией является FIND-SET и что путь поиска содержит s узлов. Фактическая стоимость операции FIND-SET составляет $O(s)$. Покажем, что отсутствуют узлы, потенциал которых возрастает вследствие операции FIND-SET, и что как минимум у $\max(0, s - (\alpha(n) + 2))$ узлов на пути поиска потенциал уменьшается по меньшей мере на 1.

Чтобы увидеть отсутствие узлов с возрастающим потенциалом, обратимся к лемме 21.10 для всех узлов, не являющихся корнем. Если же узел x является корнем, то его потенциал равен $\alpha(n) \cdot x.rank$ и остается неизменным.

Теперь покажем, что потенциал как минимум $\max(0, s - (\alpha(n) + 2))$ узлов уменьшается по меньшей мере на 1. Пусть x — узел на пути поиска, такой, что $x.rank > 0$, и за x на пути поиска следует другой узел y, не являющийся корнем, где непосредственно перед выполнением операции FIND-SET $\text{level}(y) = \text{level}(x)$ (узел y не обязательно следует *непосредственно* за x). Этим ограничениям на x удовлетворяют все узлы на пути поиска, кроме не более чем $\alpha(n) + 2$ узлов. Приведенным условиям не удовлетворяют первый узел на пути поиска (если он имеет нулевой ранг), последний узел пути (т.е. корень), а также последний узел w на пути, для которого $\text{level}(w) = k$ для каждого $k = 0, 1, 2, \ldots, \alpha(n) - 1$.

Зафиксировав такой узел x, покажем, что потенциал x уменьшается как минимум на 1. Пусть $k = \text{level}(x) = \text{level}(y)$. Непосредственно перед сжатием пути в процедуре FIND-SET мы имеем

$$\begin{aligned}
x.p.rank &\geq A_k^{(\text{iter}(x))}(x.rank) && \text{(по определению iter}(x))\ , \\
y.p.rank &\geq A_k(y.rank) && \text{(по определению level}(y))\ , \\
y.rank &\geq x.p.rank && \text{(согласно следствию 21.5 и поскольку} \\
& && y \text{ следует за } x \text{ на пути поиска)}\ .
\end{aligned}$$

Объединив приведенные неравенства и обозначив через i значение $\text{iter}(x)$ перед сжатием пути, получаем

$$\begin{aligned}
y.p.rank &\geq A_k(y.rank) \\
&\geq A_k(x.p.rank) && \text{(поскольку } A_k(j) \text{ строго возрастающая)} \\
&\geq A_k(A_k^{(\text{iter}(x))}(x.rank)) \\
&= A_k^{(i+1)}(x.rank)\ .
\end{aligned}$$

Поскольку после сжатия пути и x, и y имеют один и тот же родительский узел, мы знаем, что после сжатия пути $x.p.rank = y.p.rank$ и что сжатие пути не уменьшает $y.p.rank$. Поскольку $x.rank$ не изменяется, после сжатия пути $x.p.rank \geq A_k^{(i+1)}(x.rank)$. Таким образом, сжатие пути приводит к тому, что либо увеличивается $\text{iter}(x)$ (как минимум до $i+1$), либо увеличивается $\text{level}(x)$ (что происходит, когда $\text{iter}(x)$ увеличивается как минимум до $x.rank + 1$). В любом случае в соответствии с леммой 21.10 $\phi_q(x) \leq \phi_{q-1}(x) - 1$. Следовательно, потенциал x уменьшается как минимум на 1.

Амортизированная стоимость операции FIND-SET равна фактической стоимости плюс изменение потенциала. Фактическая стоимость равна $O(s)$, и мы показали, что общий потенциал уменьшается как минимум на $\max(0, s - (\alpha(n) + 2))$. Амортизированная стоимость, следовательно, не превышает $O(s) - (s - (\alpha(n) + 2)) = O(s) - s + O(\alpha(n)) = O(\alpha(n))$, так как мы можем масштабировать потенциал таким образом, чтобы можно было пренебречь константой, скрытой в $O(s)$. ■

Теперь, после того как мы доказали все приведенные выше леммы, перейдем к следующей теореме.

Теорема 21.14

Последовательность из m операций MAKE-SET, UNION и FIND-SET, n из которых — операции MAKE-SET, может быть выполнена над лесом непересекающихся множеств с использованием объединения по рангу и сжатия пути за время $O(m\,\alpha(n))$ в наихудшем случае.

Доказательство. Непосредственно следует из лемм 21.7 и 21.11–21.13. ■

Упражнения

21.4.1

Докажите лемму 21.4.

21.4.2

Докажите, что каждый узел имеет ранг, не превышающий $\lfloor \lg n \rfloor$.

21.4.3

Сколько в свете упр. 21.4.2 битов требуется для хранения $x.rank$ для каждого узла x?

21.4.4

Используя решение упр. 21.4.2, приведите простое доказательство того факта, что операции над лесом непересекающихся множеств с использованием объединения по рангу, но без сжатия пути, выполняются за время $O(m \lg n)$.

21.4.5

Профессор полагает, что поскольку ранг узла строго возрастает вдоль простого пути к корню, уровни узлов должны монотонно возрастать вдоль этого пути.