Тема 4. Методы поиска

4.2. Логарифмический поиск в динамических таблицах

4.2.1. АВЛ-деревья

Для динамических таблиц последовательность операций включения и исключения носит случайный характер. В результате порождаются случайные деревья бинарного поиска. Если алгоритмом включения последовательно включаются n имен в первоначально пустое дерево и если все перестановки входной последовательности имен равновероятны, то среднее время поиска в таких деревьях равно $(2 \ln 2) \log n + O(1)$, т. е. примерно в 1,4 раза больше, чем среднее время поиска в полностью сбалансированном дереве с n вершинами, равное $\log n + O(1)$. Для многих приложений увеличение времени поиска примерно на 40 % можно считать вполне допустимым, однако ДБП без ограничений на практике не надежны. В худшем случае могут порождаться вырожденные ДБП, являющиеся, по существу, линейными списками, в которых время поиска становится линейным относительно величины таблицы, а не логарифмическим. Поэтому в ряде приложений необходимы специальные методы балансировки, гарантирующие логарифмическое время поиска даже в худшем случае без требований каких-либо допущений о последовательности операций включения и исключения.

Наиболее очевидный путь для того, чтобы динамическое дерево не стало асимметричным, – поддерживать его полностью сбалансированным во все моменты времени. Тогда если при включении в дерево новой вершины с именем *C* дерево становится не полностью сбалансированным, то оно должно быть перестроено (рис. 4.5). К сожалению, такое преобразование обычно меняет все дерево, и ни одно из отношений отец-сын не остается неизменным, т. е. такая операция перестройки в общем случае требует времени, пропорционального числу вершин.

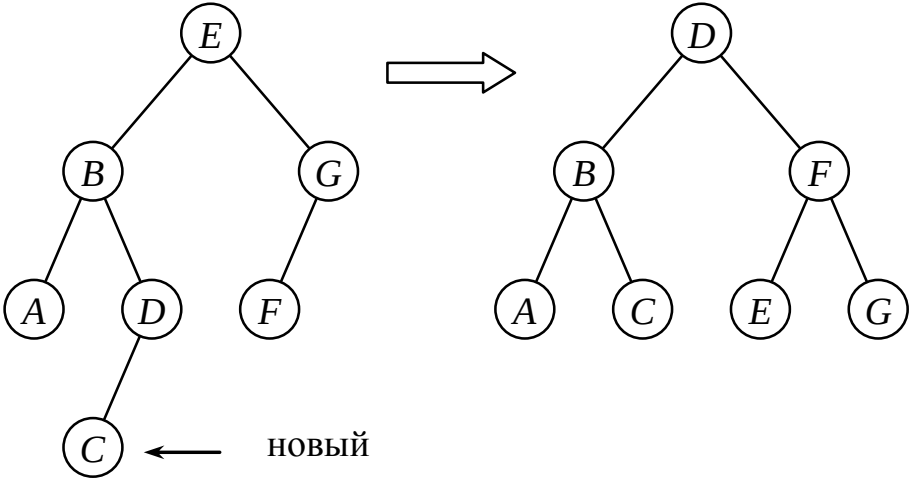

Рис. 4.5. Восстановление полной сбалансированности бинарного дерева

Поэтому необходимы такие методы балансировки, которые после выполнения операций включения и исключения позволяют преобразовать дерево локальными изменениями вдоль одного пути от корня к листу, что требует времени $O(\log n)$. Для обеспечения такой гибкости деревья должны иметь возможность отклоняться от полностью сбалансированных деревьев, но в такой малой степени, что среднее время поиска в них было бы лишь немногим больше, чем в полностью сбалансированном бинарном дереве.

Пусть $h(T)$ обозначает высоту бинарного дерева T, т. е. длину самого длинного пути от корня к листу. Высота дерева с единственной вершиной равна 0, для удобства высоту пустого дерева считают равной -1.

Дерево бинарного поиска называется *АВЛ-деревом* (в честь его авторов Г.М. Адельсона-Вельского и Е.М. Ландиса), или *сбалансированным по высоте*, если два поддерева корня T_l (левое) и T_r (правое) удовлетворяют следующим условиям:

1. $|h(T_l) - h(T_r)| \leq 1$,
2. T_l и T_r сбалансированы по высоте.

Таким образом, в АВЛ-дереве для каждой вершины высоты ее левого и правого поддеревьев отличаются не более чем на единицу. Пустое дерево, не имеющее ни корня, ни поддеревьев, удовлетворяет этим условиям и, следовательно, является АВЛ-деревом. Дерево с единственной вершиной также является АВЛ-деревом. Пример АВЛ-дерева показан на рис. 4.6, *а*. Дерево на рис. 4.6, *б* не является АВЛ-деревом, поскольку высота левого (пустого) поддерева вершины *D* равна –1, а правого – единице, т. е. высоты поддеревьев отличаются на 2.

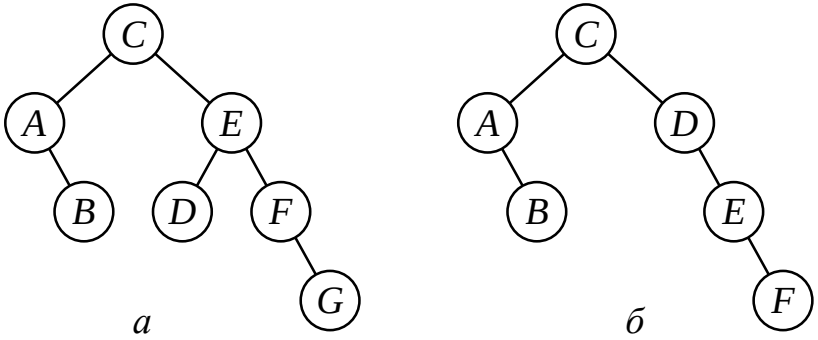

Рис. 4.6. Деревья бинарного поиска: *а* – АВЛ-дерево; *б* – не АВЛ-дерево

Высота наиболее асимметричного АВЛ-дерева с n вершинами составляет примерно $1{,}44 \log n$, т. е. время поиска в худшем случае приблизительно на 44 % больше, чем среднее время поиска в полностью сбалансированных бинарных деревьях. Среднее же время поиска в наиболее асимметричных АВЛ-деревьях примерно на 4 % больше. Учитывая тот факт, что сильно асимметричной является только малая часть АВЛ-деревьев, среднее время поиска в АВЛ-деревьях, усредненное по всем таким деревьям, равно $\log n + O(1)$.

В результате выполнения операции включения или исключения сбалансированность дерева может быть нарушена. Основной операцией, изменяющей структуру дерева для восстановления сбалансированности, является операция *вращения*, которая представляет собой локальную операцию и сохраняет свойство упорядоченности вершин дерева (имен в таблице, представленной АВЛ-деревом) в соответствии с симметричным прохождением. На рис. 4.7 показаны два взаимно обратных вращения: левое и правое (α, β и γ обозначают произвольные поддеревья). В результате правого вращения вершин *A* и *B* (правое вращение вокруг вершины *B*) вершина *B* становится правым сыном вершины *A*, а бывший правый сын вершины *A* (поддерево β) становится левым сыном вершины *B*. Таким образом, вершина *A* с левым поддеревом α поднимается на один уровень вверх, а вершина *B* с правым поддеревом γ опускается на один уровень вниз, поддерево β остается на том же уровне. Аналогично (в зеркальном отражении) работает левое вращение. Операции вращения выполняются за время $O(1)$ и заключаются в изменении нескольких указателей.

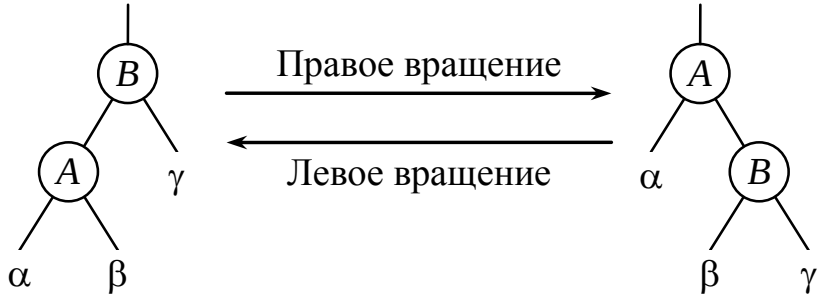

Рис. 4.7. Операции вращения

В ряде случаев для восстановления сбалансированности дерева возникает необходимость в последовательном выполнении двух вращений, которые для удобства можно объединить в одну операцию, называемую *двойным вращением*. На рис. 4.8 показано двойное вращение (α, β, γ и δ произвольные поддеревья), когда сначала выполняется правое вращение вокруг вершины *C*, а затем левое вращение вокруг вершины *A*. В результате двойного вращения вершина *A* с левым поддеревом α опускается на один уровень вниз, вершина *B* поднимается на два уровня вверх, уровень вершины *C* с правым поддеревом δ остается прежним, поддеревья β и γ поднимаются на один уровень вверх. Имеется также симметричный вариант двойного вращения.

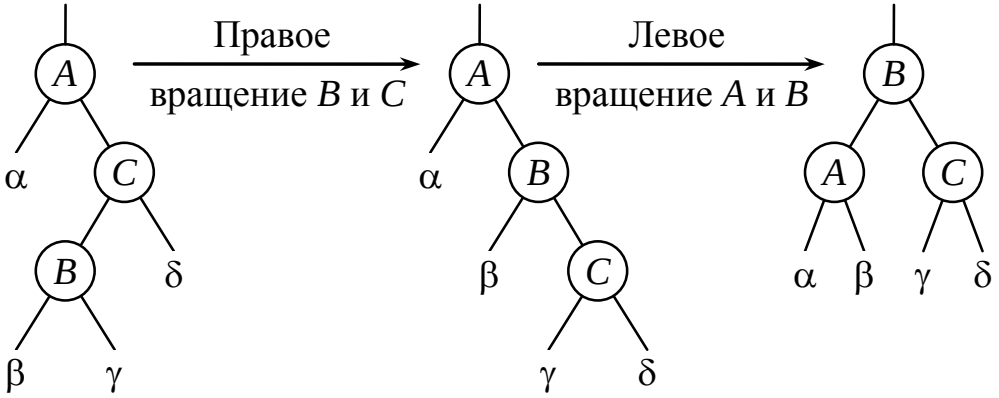

Рис. 4.8. Двойное вращение

Для контроля требований сбалансированности по высоте в процессе выполнения операций включения и исключения необходима информация о соотношении высот поддеревьев каждой вершины. Чтобы не рассматривать не затронутые локальными преобразованиями поддеревья для определения их высот, лучше всего иметь такую информацию (назовем ее *балансом*) в каждой вершине дерева. При узловом представлении дерева в структуру узла следует добавить специальное поле для хранения баланса.

Баланс некоторой вершины дерева определяется как разность высот левого и правого поддеревьев. В АВЛ-деревьях, поскольку высоты поддеревьев каждой вершины отличаются не более чем на единицу, баланс вершины может иметь только три значения: 1, 0 или –1 в зависимости от того, что высота ее левого поддерева больше, равна или меньше высоты правого поддерева. На рисунках баланс будем записывать внутри вершин.

Включение новой вершины. Для включения новой вершины *new* применяется обычный алгоритм включения в дерево бинарного поиска. После включения вершины путь из корня в лист проходится в обратном направлении. Для обеспечения возвращения необходимо модифицировать алгоритм включения в части поиска так, чтобы сохранить спускающийся путь в стеке. Другой подход предполагает добавление в структуру узла дерева специального поля *father*, указывающего на отца данной вершины. В этом случае нет необходимости в сохранении спускающегося пути в стеке. Действия, предпринимаемые при прохождении каждой вершины по восходящему пути, зависят от баланса в этой вершине и в случае, если баланс равен 1 или –1, от направления последнего шага или двух последних шагов вверх по этому пути. Возможны следующие ситуации:

1. Если баланс текущей вершины равен 0, то его значение меняется на –1, если последний шаг начинался из правого сына, и на 1, если он начинался из левого сына. Поскольку высота поддерева, корнем которого является текущая вершина, увеличивается, продолжается движение вверх по пути с информацией об увеличении высоты дерева. Процедура заканчивается, если будет достигнут корень дерева или будет обработана ситуация 2 или 3.

2. Если баланс текущей вершины равен 1 или –1 и последний шаг начинался из более короткого из двух поддеревьев текущей вершины, то значение баланса меняется на 0. Поскольку высота поддерева, корнем которого является текущая вершина, не изменяется, движение вверх прекращается и процедура заканчивается.

3. Если баланс текущей вершины равен 1 или –1 и последний шаг начинался из более высокого из двух поддеревьев текущей вершины, то:

а) если последние два шага делались в одном направлении (оба из левых сыновей или оба из правых сыновей), то выполняется соответствующее вращение; на рис. 4.9, *а* показан один из симметричных вариантов такого преобразования, когда два последних шага к вершине *A* делались из правых сыновей;

б) если два последних шага делались в противоположных направлениях, то выполняется соответствующее двойное вращение; на рис. 4.9, *б* показан один из симметричных вариантов, когда два последних шага к вершине *A* делались в противоположных направлениях.

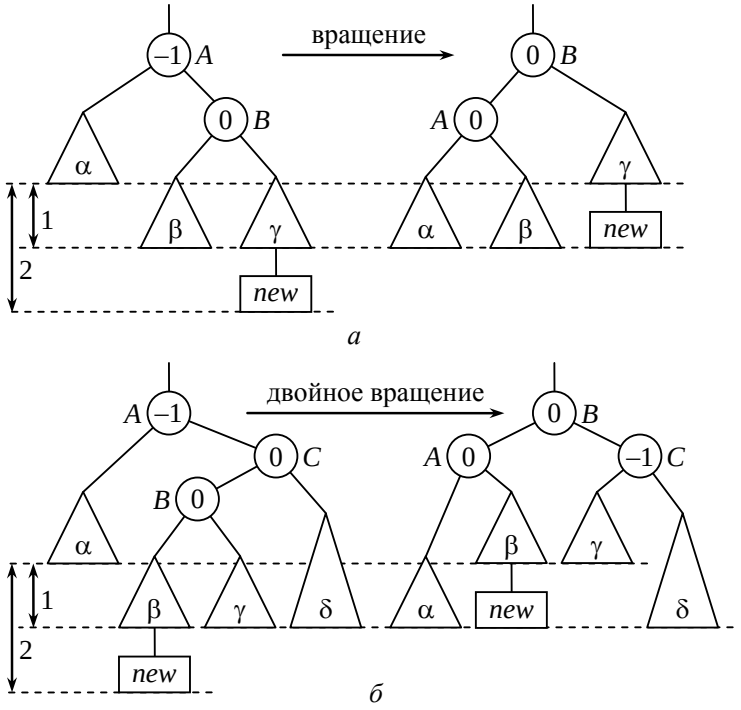

Рис. 4.9. Восстановление сбалансированности АВЛ-дерева после включения:
а – два последних шага к вершине *A* сделаны из правых сыновей;
б – два последних шага к вершине *A* сделаны сначала из левого сына (от *B* к *C*), потом из правого (от *C* к *A*)

В обоих случаях высота поддерева после балансировки остается той же, что и до включения новой вершины, т. е. эти преобразования не влияют на баланс расположенных выше вершин. Поэтому продвижение вверх прекращается и процедура заканчивается. Таким образом, при включении новой вершины может потребоваться не более одного преобразования структуры дерева (вращения или двойного вращения).

Исключение вершины. Рассмотренный ранее алгоритм исключения гарантирует то, что удаляемая вершина имеет не менее одного пустого поддерева. Если удаляемая вершина имеет ровно одно пустое поддерево, то согласно ограничению на высоту поддеревьев, другое поддерево состоит только из одной вершины. В этом случае влияние на сбалансированность дерева по высоте такое же, как если бы исключался лист. Когда лист исключается, поддерево, состоящее именно из этой вершины, теряет одну единицу высоты. Необходимо пройти путь от этого исключенного листа вверх до корня дерева, проверяя в каждой вершине, какое влияние на большее дерево оказывает уменьшение высоты поддерева. Действия, предпринимаемые в каждой вершине, расположенной на восходящем пути, зависят от баланса текущей вершины, направления последнего шага и в некоторых случаях – от баланса сына текущей вершины. Так же, как и при включении, в ряде случаев преобразование восстанавливает высоту поддерева до той, которая была до исключения (процедура заканчивается); в других случаях продолжается движение вверх по пути к корню с информацией об уменьшении высоты поддерева. Возможны следующие ситуации:

1. Если баланс текущей вершины равен 0, то уменьшение высоты любого ее поддерева не влияет на высоту поддерева, имеющего корнем текущую вершину. Изменяется значение баланса в зависимости от того, высота какого поддерева уменьшилась (на 1 или –1). Процедура заканчивается.

2. Если баланс текущей вершины равен 1 или –1 и последний шаг начинался из более высокого из двух поддеревьев, то значение баланса меняется на 0 и продолжается движение вверх с информацией об уменьшении высоты поддерева.

3. Если баланс текущей вершины равен 1 или –1 и последний шаг начинался из более короткого из двух поддеревьев, то в текущей вершине нарушается соотношение высот поддеревьев. Различаются три подслучая и их симметричные варианты в соответствии с балансом другого сына текущей вершины. На рис. 4.10 представлены варианты для ситуации, когда произошло уменьшение высоты левого поддерева α текущей вершины *A* (показано пунктирными линиями). Следовательно, для распознавания подслучаев необходимо анализировать баланс правого сына текущей вершины.

а) если баланс правого сына текущей вершины равен 0 (рис. 4.10, *а*), то выполняется соответствующее вращение, которое восстанавливает соотношение высот поддеревьев в текущей вершине без изменения высоты поддерева, имеющего корнем текущую вершину. Процедура заканчивается;

б) если баланс правого сына текущей вершины равен –1 (рис. 4.10, *б*), то выполняется соответствующее вращение, которое восстанавливает соотношение высот поддеревьев в текущей вершине. Продолжается движение вверх с информацией об уменьшении высоты поддерева, корнем которого является текущая вершина;

в) если баланс правого сына текущей вершины равен 1, (рис. 4.10, *в*), то выполняется соответствующее двойное вращение, которое восстанавливает соотношение высот поддеревьев в текущей вершине. Продолжается движение вверх с информацией об уменьшении высоты поддерева, корнем которого является текущая вершина. На рис. 4.10, *в* баланс вершины *B* может иметь любое из значений 1, 0 или –1, но, по крайней мере, одно из двух поддеревьев вершины *B* должно иметь высоту, обозначенную закрашенной областью. Балансы вершин *A* и *C* после преобразования зависят от высоты двух закрашенных поддеревьев.

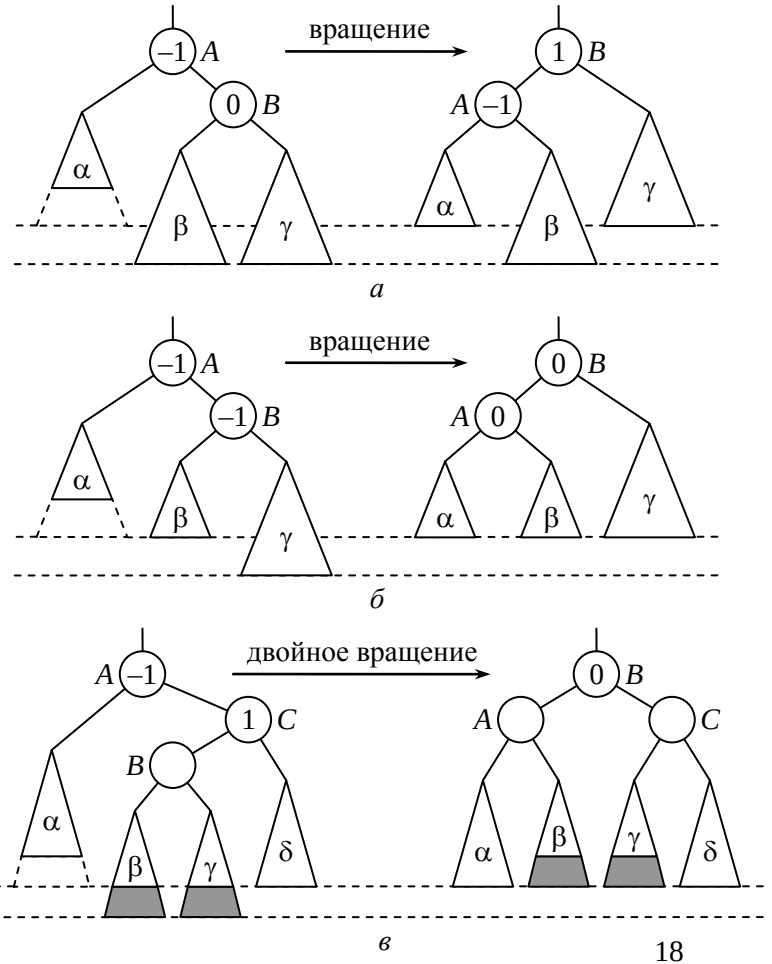

Рис. 4.10. Восстановление сбалансированности АВЛ-дерева после исключения:
а – баланс правого сына *B* текущей вершины *A* равен 0;
б – баланс правого сына *B* текущей вершины *A* равен –1;
в – баланс правого сына *B* текущей вершины *A* равен 1

Следует обратить внимание на то, что в противоположность операции включения, требующей не более одного преобразования, исключение может требовать в худшем случае $\lfloor h/2 \rfloor$ преобразований, где h – высота всего дерева. Однако в большинстве случаев процедура исключения не потребует прослеживания восходящего пути до самого корня. Обычно она завершается после некоторого числа шагов, не зависящего от высоты дерева.