









Node \*AVL::rotate(Node \*a) {  
 *int* d = a->diff;  
 *if* (d == -2) {Node \*b = a->r;  
 *if* (b->diff == 1) {  
 Node \*c = b->l;  
 *if* (c->diff == 1)  
 a->diff = 0, b->diff = -1, c->diff = 0;  
 *else if* (c->diff == -1)  
 a->diff = 1, b->diff = 0, c->diff = 0;  
 *else* a->diff = 0, b->diff = 0, c->diff = 0;  
 a = big\_rotate\_left(a);  
 }  
 *else* {  
 *if* (b->diff == -1)  
 b->diff = 0, a->diff = 0;  
 *else* b->diff = 1, a->diff = -1;  
 a = rotate\_left(a);  
 }  
 } *else if* (d == 2) {Node \*b = a->l;  
 *if* (b->diff == -1) {  
 Node \*c = b->r;  
 *if* (c->diff == -1)  
 a->diff = 0, b->diff = 1, c->diff = 0;  
 *else if* (c->diff == 1)  
 a->diff = -1, b->diff = 0, c->diff = 0;  
 *else* a->diff = 0, b->diff = 0, c->diff = 0;  
 a = big\_rotate\_right(a);  
 }  
 *else* {  
 *if* (b->diff == 0)  
 b->diff = -1, a->diff = 1;  
 *else* b->diff = 0, a->diff = 0;  
 a = rotate\_right(a);  
 }  
 }  
 *return* a;  
}

Node \*AVL::rotate\_left(Node \*a) {  
 Node \*b = a->r;  
 a->r = b->l;  
 b->l = a;  
 *return* b;  
}  
  
Node \*AVL::rotate\_right(Node \*a) {  
 Node \*b = a->l;  
 a->l = b->r;  
 b->r = a;  
 *return* b;  
}  
  
Node \*AVL::big\_rotate\_left(Node \*a) {  
 a->r = rotate\_right(a->r);  
 a = rotate\_left(a);  
 *return* a;  
}  
  
Node \*AVL::big\_rotate\_right(Node \*a) {  
 a->l = rotate\_left(a->l);  
 a = rotate\_right(a);  
 *return* a;  
}

**Добавление вершины**

Пусть нам надо добавить ключ tt. Будем спускаться по дереву, как при поиске ключа tt. Если мы стоим в вершине aa и нам надо идти в поддерево, которого нет, то делаем ключ tt листом, а вершину aa его корнем. Дальше поднимаемся вверх по пути поиска и пересчитываем баланс у вершин. Если мы поднялись в вершину ii из левого поддерева, то diff[i]diff[i] увеличивается на единицу, если из правого, то уменьшается на единицу. **Если пришли в вершину и её баланс стал равным нулю, то это значит высота поддерева не изменилась и подъём останавливается. Если пришли в вершину и её баланс стал равным 11 или −1−1, то это значит высота поддерева изменилась и подъём продолжается. Если пришли в вершину и её баланс стал равным 22 или −2−2, то делаем одно из четырёх вращений и, если после вращения баланс стал равным нулю, то останавливаемся, иначе продолжаем подъём.**

### Удаление вершины

Для простоты опишем рекурсивный алгоритм удаления. Если вершина — лист, то [удалим](https://neerc.ifmo.ru/wiki/index.php?title=%D0%94%D0%B5%D1%80%D0%B5%D0%B2%D0%BE_%D0%BF%D0%BE%D0%B8%D1%81%D0%BA%D0%B0,_%D0%BD%D0%B0%D0%B8%D0%B2%D0%BD%D0%B0%D1%8F_%D1%80%D0%B5%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F#.D0.A3.D0.B4.D0.B0.D0.BB.D0.B5.D0.BD.D0.B8.D0.B5) её, иначе найдём самую близкую по значению вершину aa, переместим её на место удаляемой вершины и [удалим](https://neerc.ifmo.ru/wiki/index.php?title=%D0%94%D0%B5%D1%80%D0%B5%D0%B2%D0%BE_%D0%BF%D0%BE%D0%B8%D1%81%D0%BA%D0%B0,_%D0%BD%D0%B0%D0%B8%D0%B2%D0%BD%D0%B0%D1%8F_%D1%80%D0%B5%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F#.D0.A3.D0.B4.D0.B0.D0.BB.D0.B5.D0.BD.D0.B8.D0.B5) вершину aa. От удалённой вершины будем подниматься вверх к корню и пересчитывать баланс у вершин. Если мы поднялись в вершину ii из левого поддерева, то diff[i]diff[i] уменьшается на единицу, если из правого, то увеличивается на единицу. **Если пришли в вершину и её баланс стал равным 11 или −1−1, то это значит, что высота этого поддерева не изменилась и подъём можно остановить. Если баланс вершины стал равным нулю, то высота поддерева уменьшилась и подъём нужно продолжить. Если баланс стал равным 22 или −2−2, следует выполнить одно из четырёх вращений и, если после вращений баланс вершины стал равным нулю, то подъём продолжается, иначе останавливается.**