

Удаление вершин из СДП

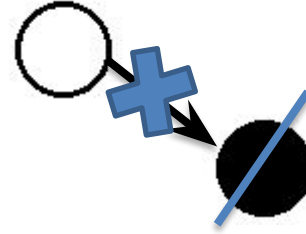
Идея:

сначала нужно найти вершину с ключом X , двигаясь влево или вправо по пути поиска, пока не остановимся на вершине с ключом X или пока не достигнем листовой вершины с нулевыми указателями.

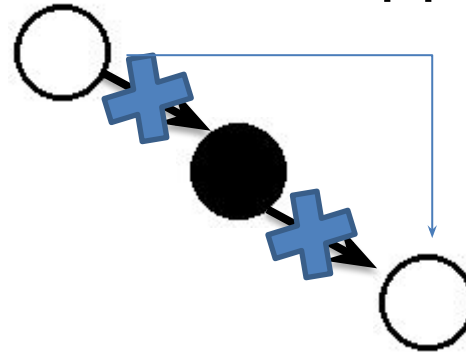
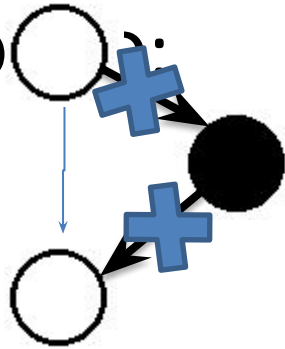
Поиск в дереве не осуществляется перебором (обходом).

Обходом осуществляется лишь распечатка вершин.

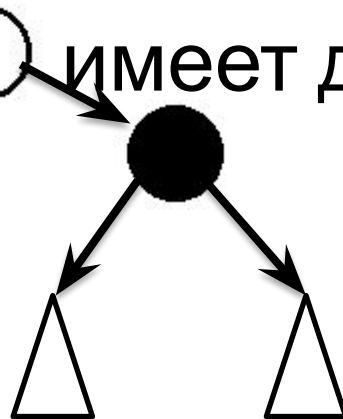
1. Если удаляемая вершина не имеет поддеревьев:

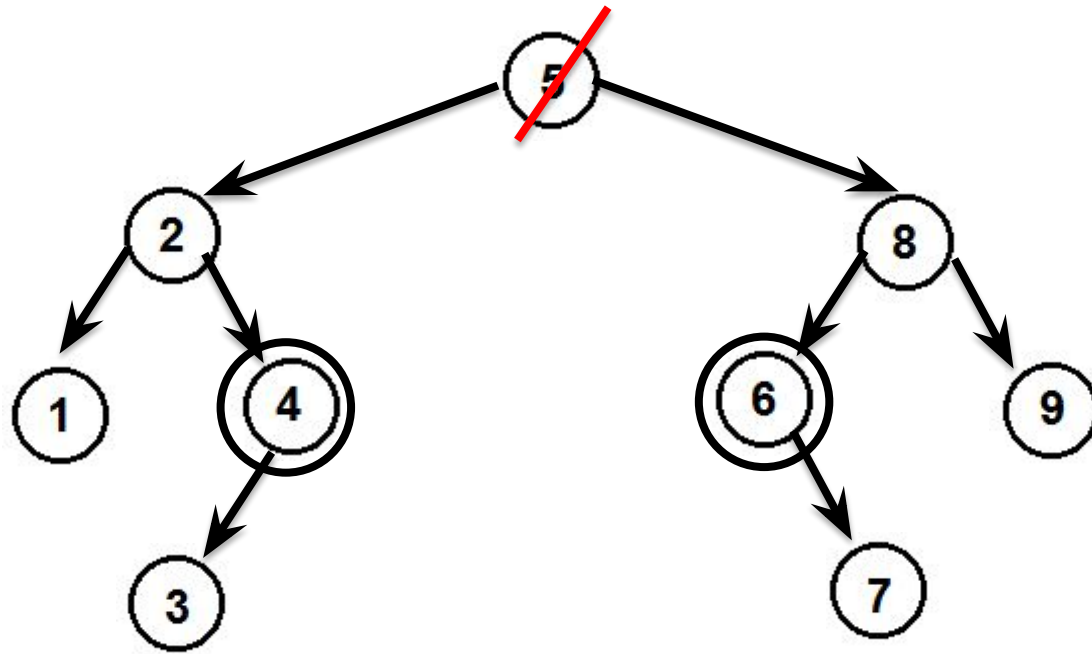


2. Если на удаляемой вершине одно поддер...



3. Если вершина имеет два поддерева:





Из этого дерева нужно удалить вершину 5, на её место можно поставить либо 4, либо 6.

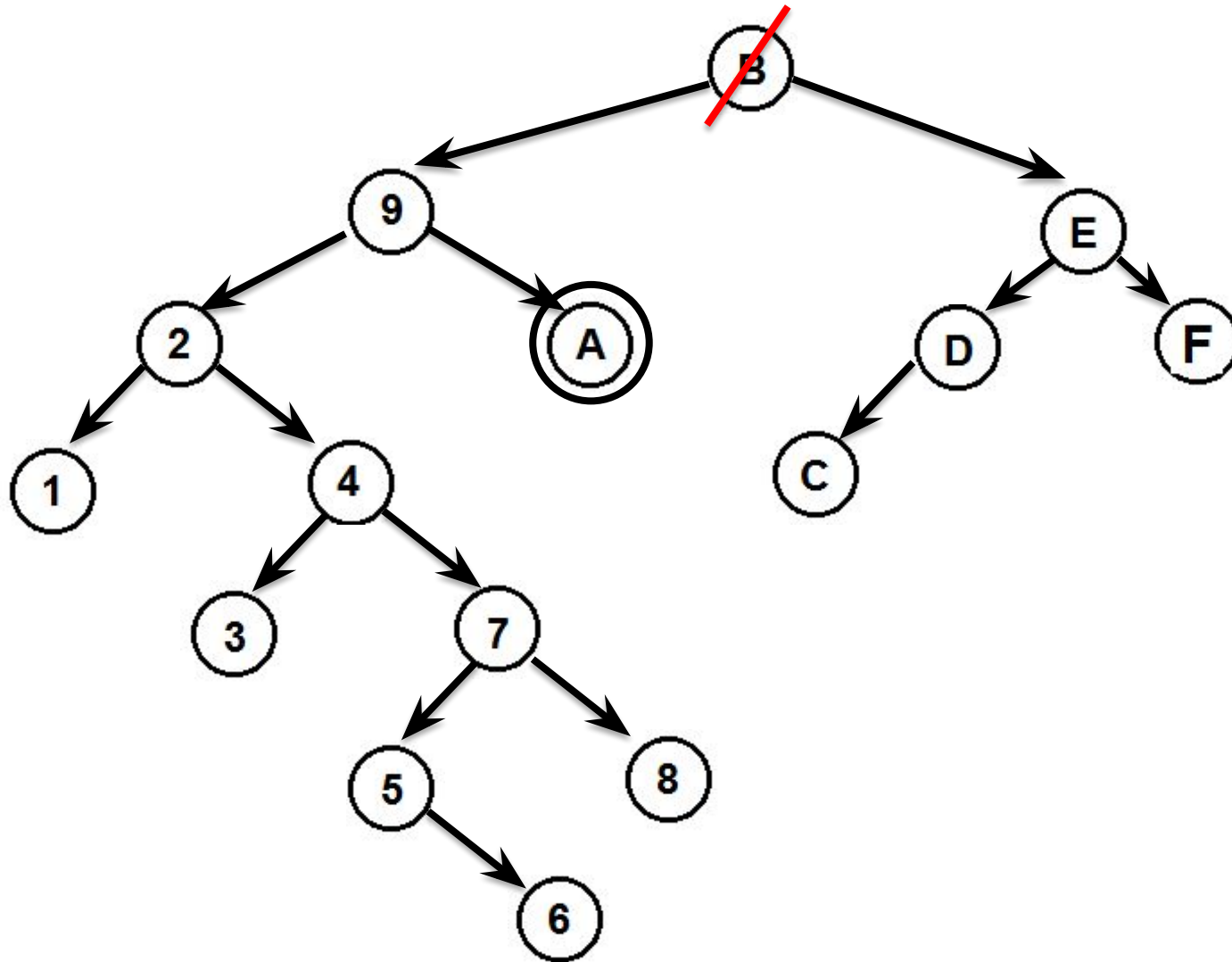
Правила удаления:

а) На место удаляемой вершины ставится наибольшая вершина из её левого поддерева, т.е. самая правая вершина из левого поддерева, не имеющая правого поддерева.

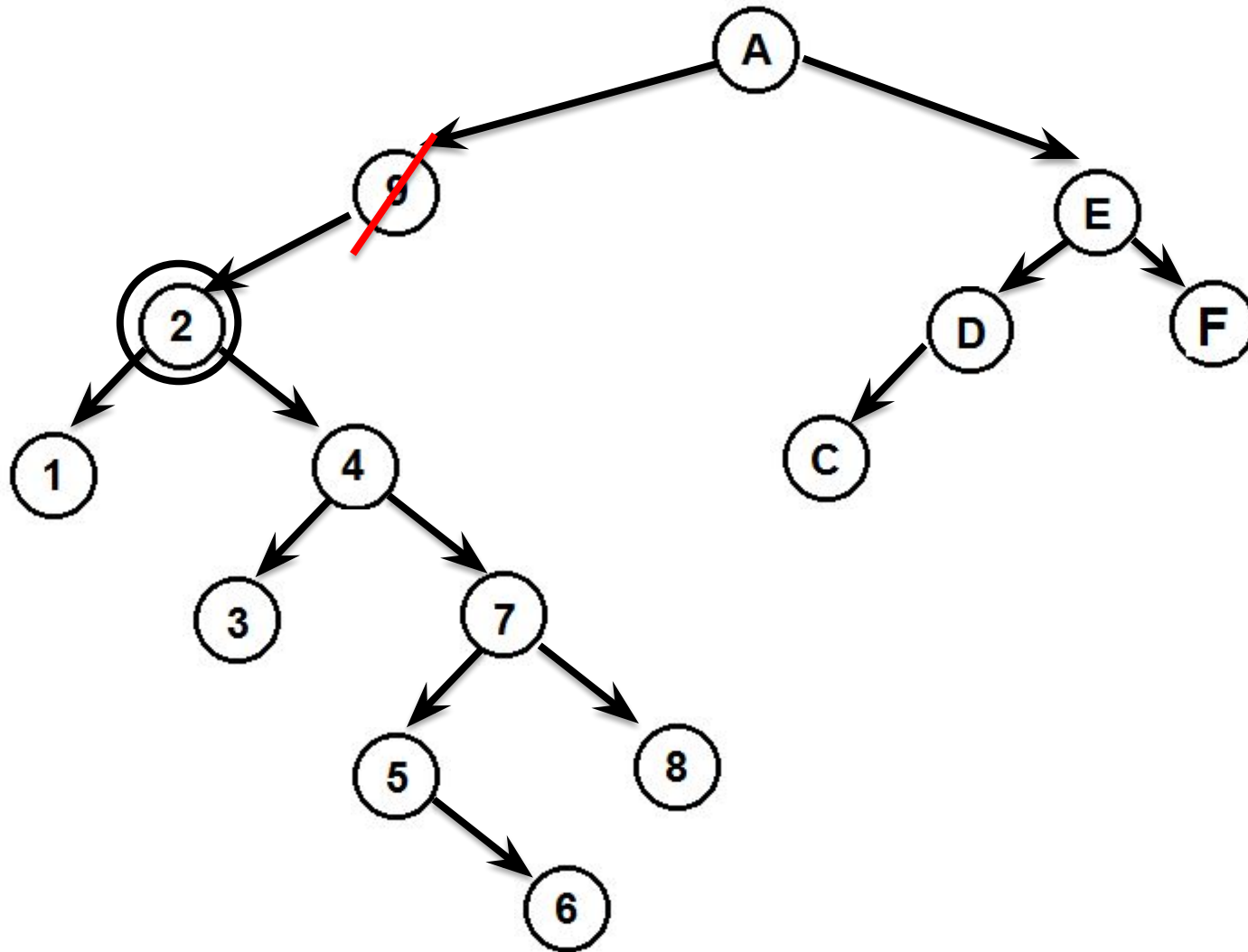
б) На место удаляемой вершины ставится наименьшая вершина из её правого поддерева, т.е. самая левая вершина из её правого поддерева, не имеющая левого поддерева.

Будем строить алгоритмы на правиле «а»

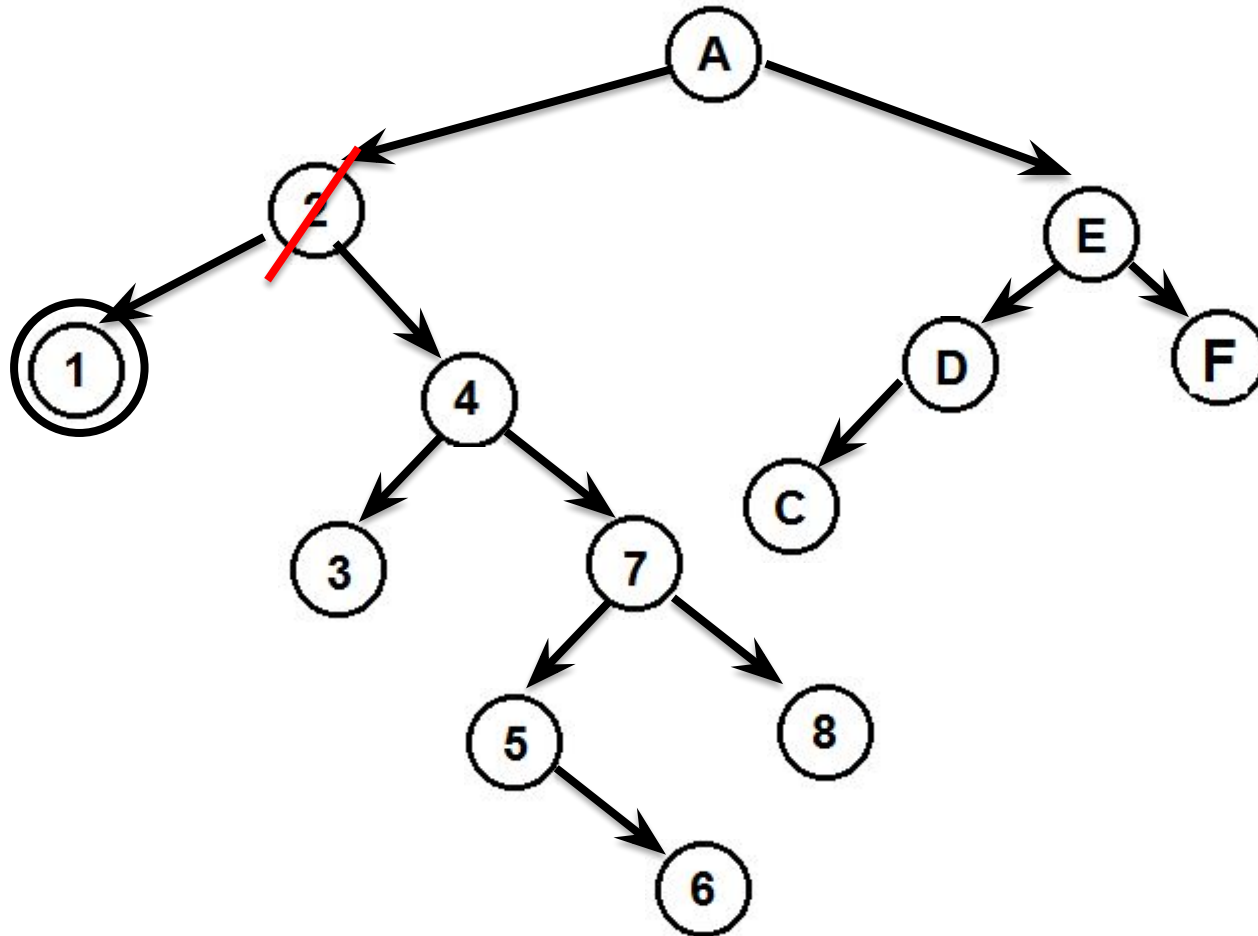
B 9 2 4 1 7 E F A D C 3 5 8 6



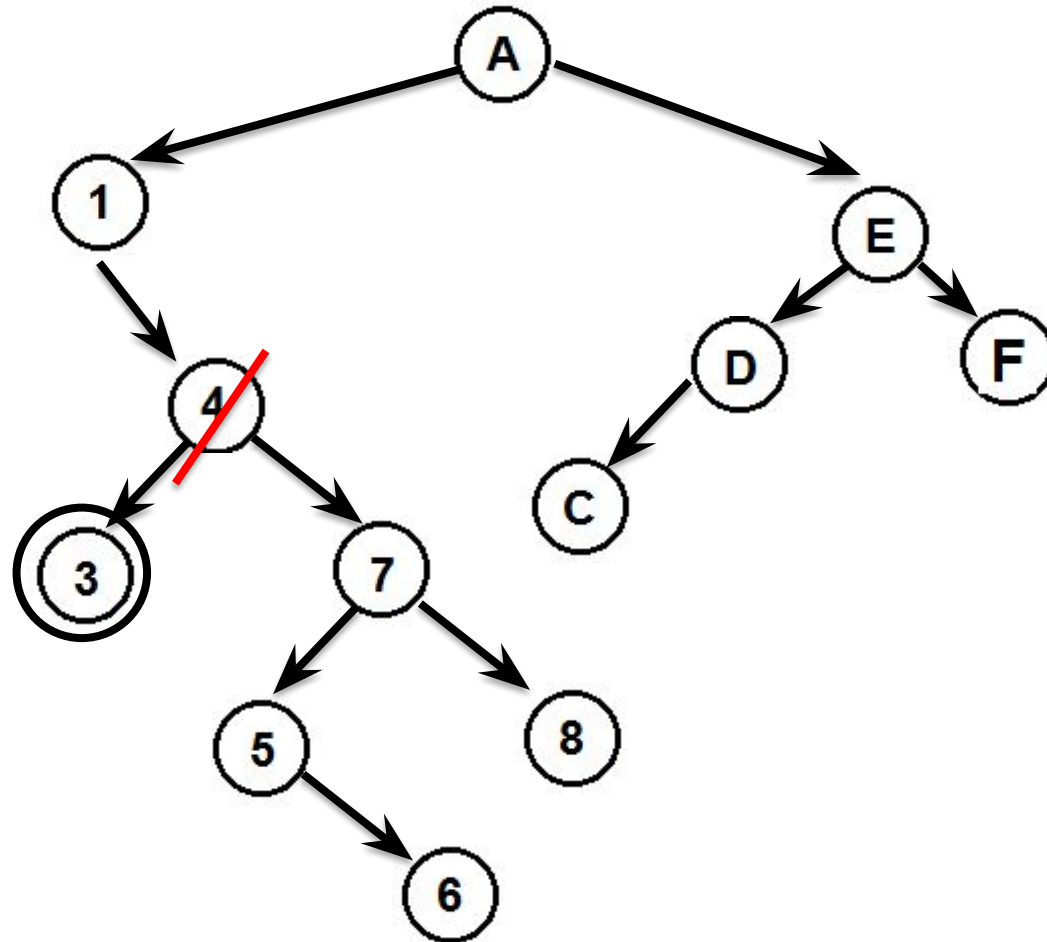
B 9 2 4 1 7 E F A D C 3 5 8 6



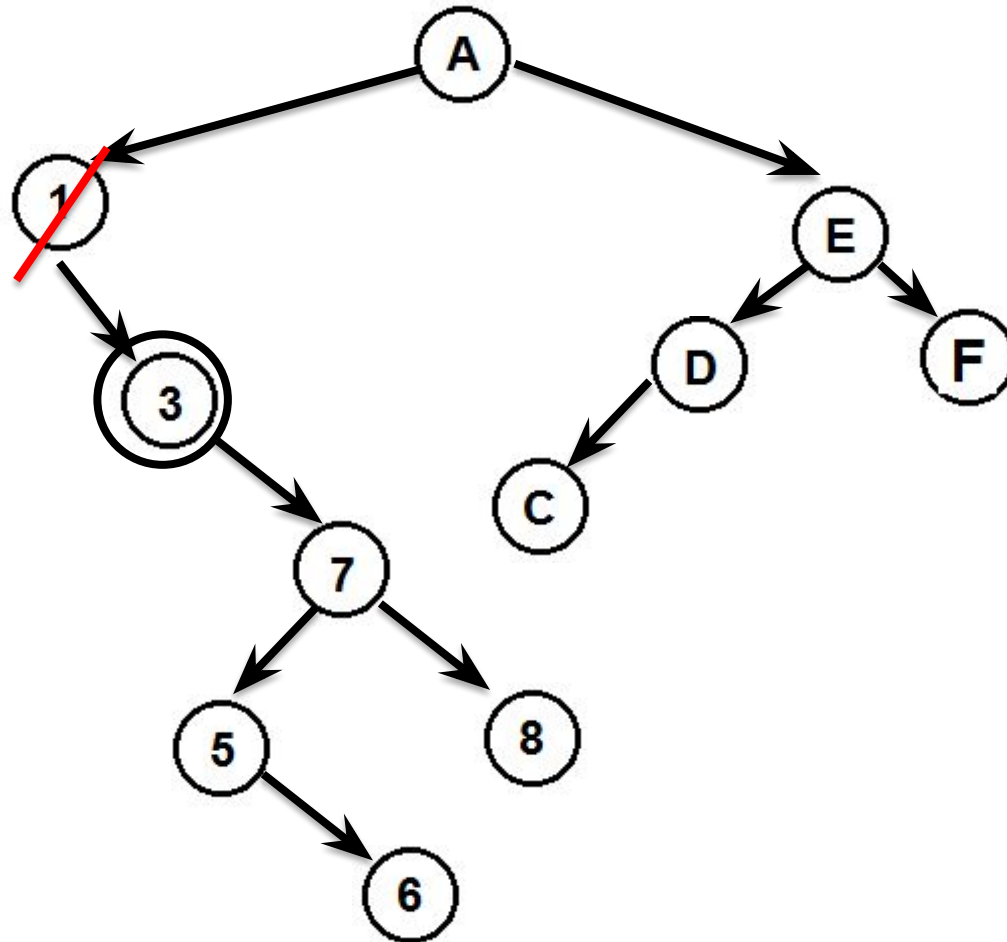
B 9 2 4 1 7 E F A D C 3 5 8 6



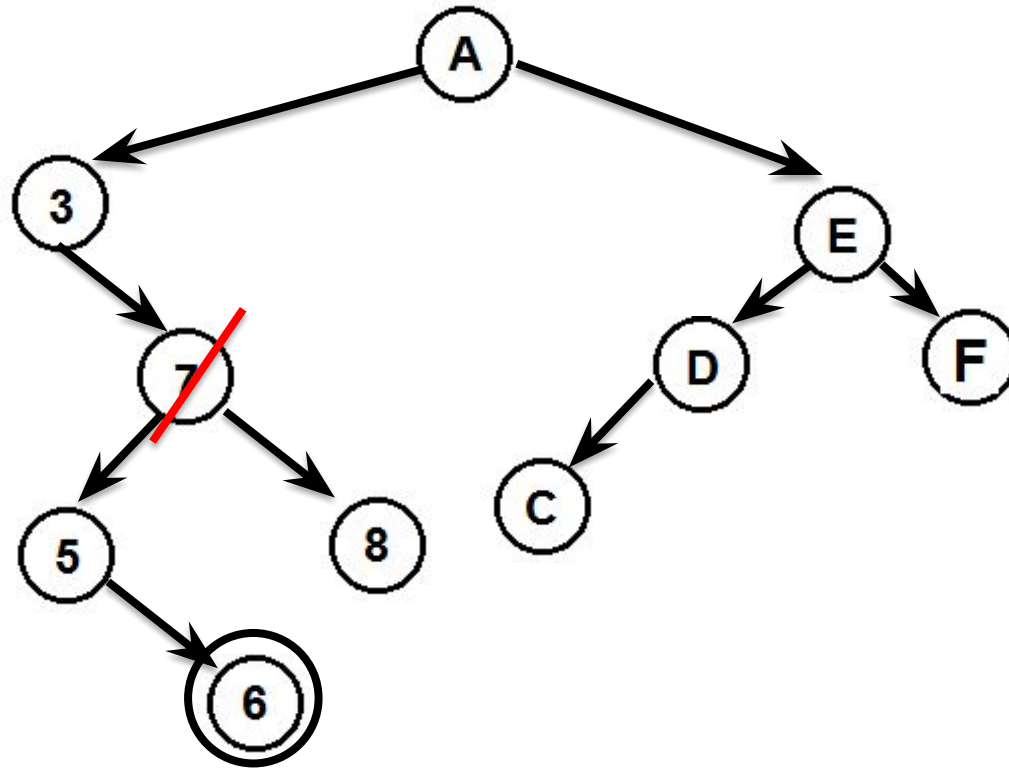
B 9 2 4 1 7 E F A D C 3 5 8 6



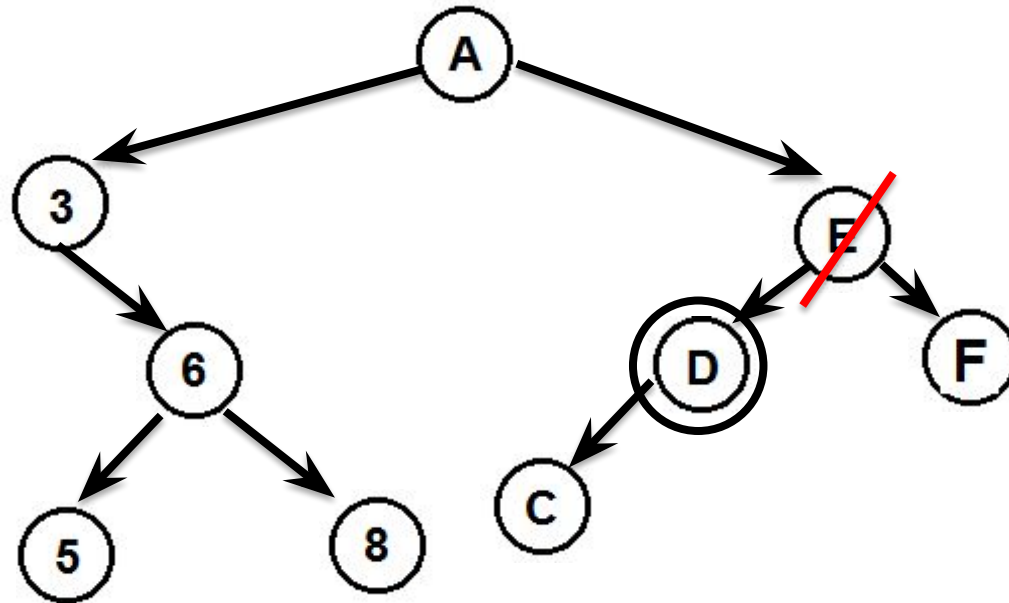
B 9 2 4 1 7 E F A D C 3 5 8 6



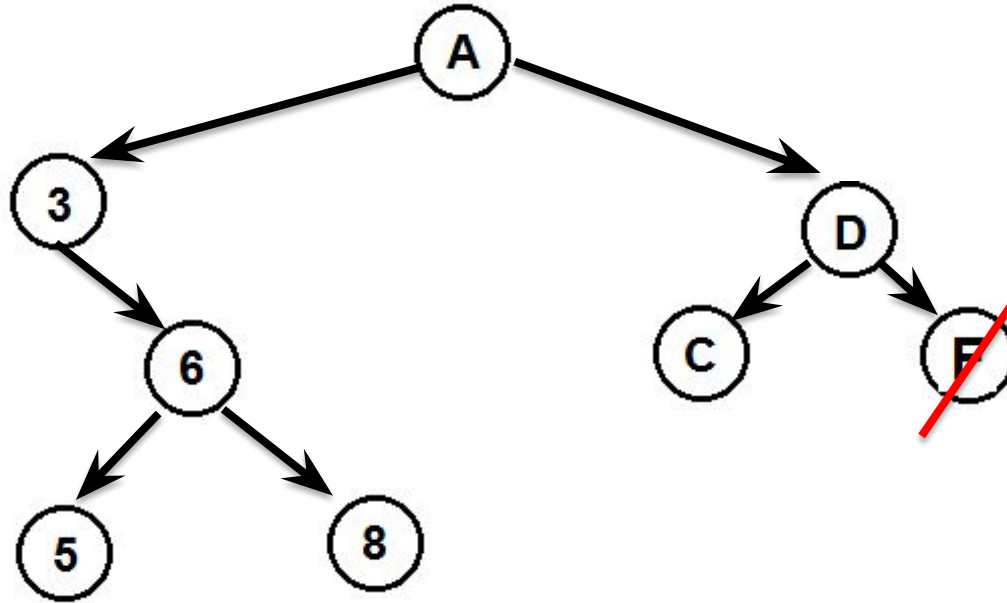
B 9 2 4 1 7 E F A D C 3 5 8 6



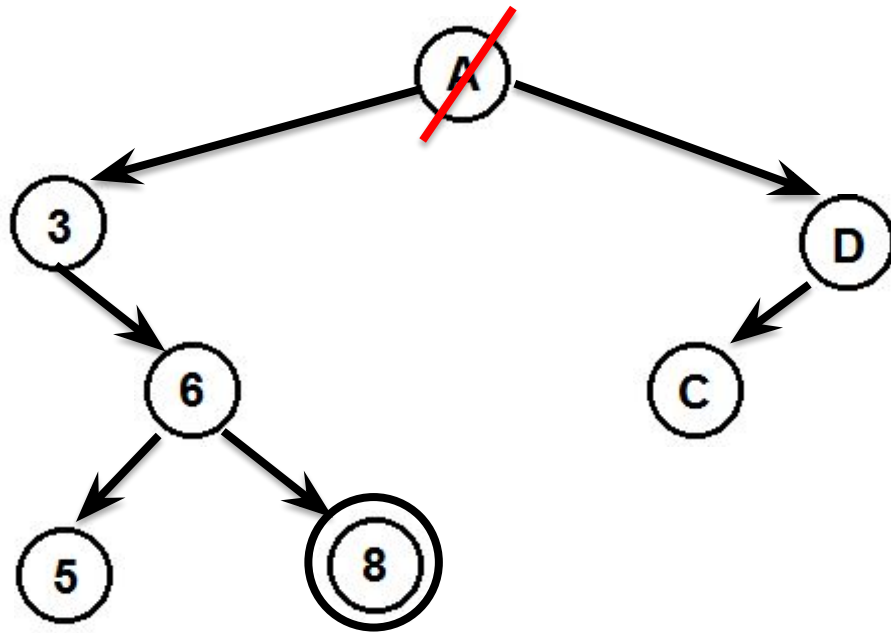
B 9 2 4 1 7 E F A D C 3 5 8 6



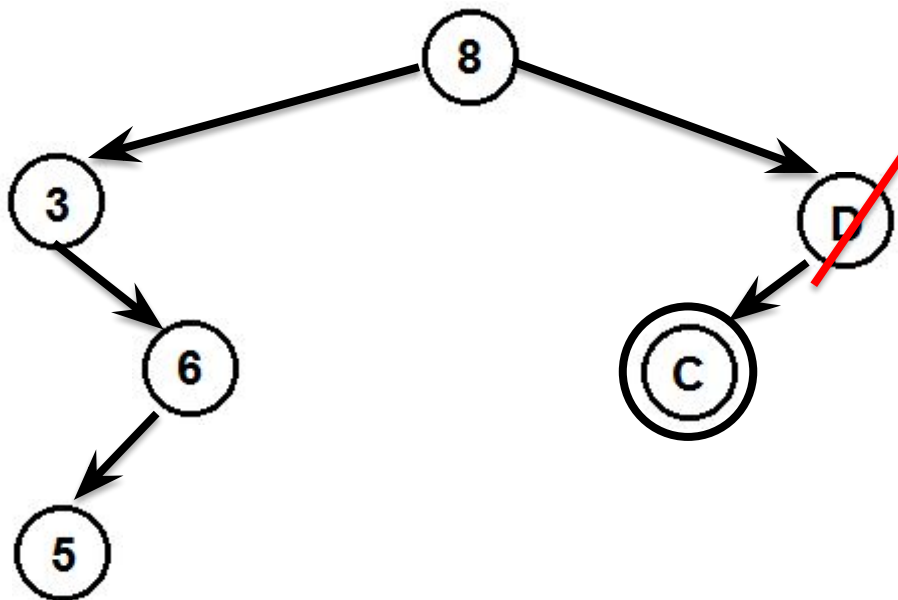
B 9 2 4 1 7 E F A D C 3 5 8 6



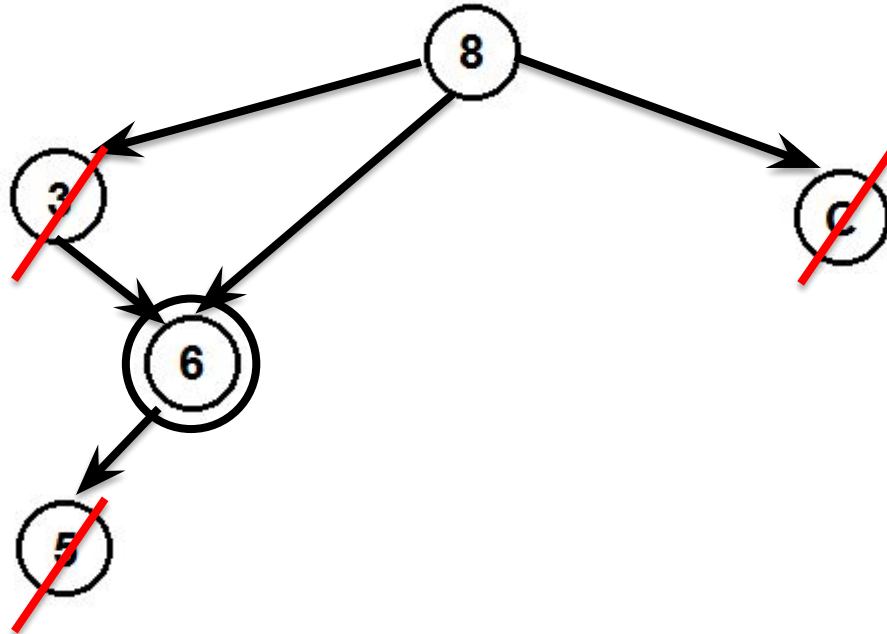
B 9 2 4 1 7 E F A D C 3 5 8 6



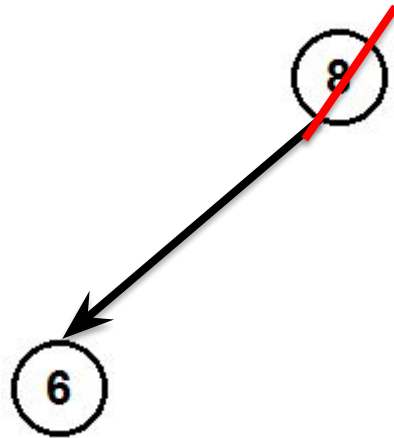
B 9 2 4 1 7 E F A D C 3 5 8 6



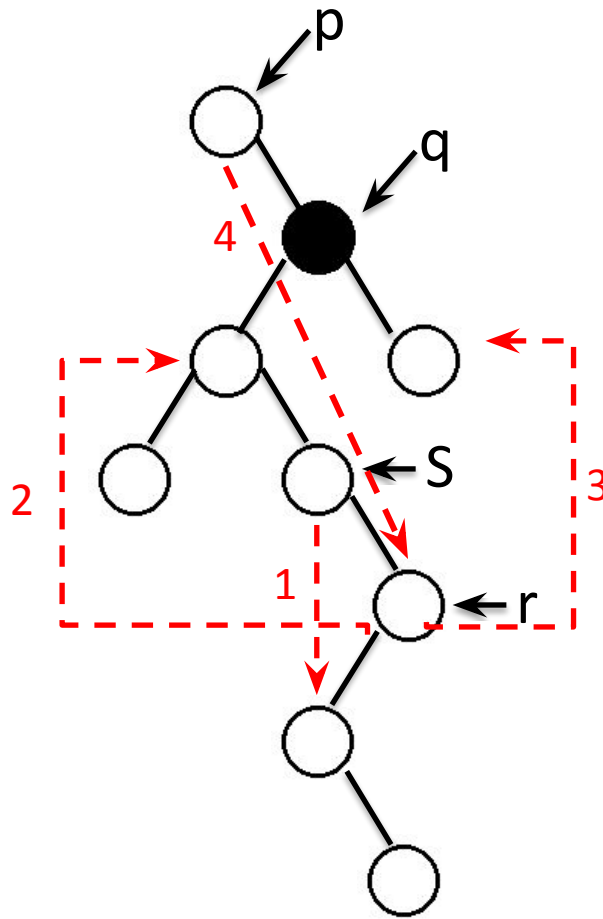
B 9 2 4 1 7 E F A D C 3 5 8 6



B 9 2 4 1 7 E F A D C 3 5 8 6



B 9 2 4 1 7 E F A D C 3 5 8 6



p - адрес адреса удаляемой вершины

q - адрес удаляемой вершины

Удаление (D, *Root)

p=&Root

```
DO (*p ≠ NULL) // поиск элемента
  IF (D<(*p)-->Data) p:=&((*p)-->Left)
  ELSE IF (D>((*p)-->Data) p:=&((*p)-->Right)
  ELSE OD {данные есть в дереве}
FI
```

OD

```
IF (*p ≠ NULL)
  q:=*p
  IF (q-->Left=NULL) *p:=q-->Right;
  ELSE IF (q-->Right=NULL) *p:=q-->Left;
  ELSE /*2 поддеревя*/
    r:=q-->Left ; S:=q;
```

IF (r-->Right = NULL)

r-->Right := q-->Right ; (3)

*p:=r; (4)

ELSE

DO (r-->Right ≠ NULL)

S:=r; r:=r-->Right ;

OD

s-->Right:=r-->Right; (1)

r-->Left:=q-->Left; (2)

r-->Right:=q-->Right; (3)

*p:=r; (4)

FI

FI

free(q)

FI

