

Brisanje čvora iz crveno-crnog binarnog stabla

SPA2

```
void obrisi(struct drvo **p,int k){
    struct drvo *x;
    x=pronadji(*p,k);
    if (x) obrisi_cc(p,x);
}
```

```
struct drvo* pronadji(struct drvo *p,int k){
    struct drvo *t=NULL;
    if (p) {
        if (p->broj==k) return p;
        if (p->broj>=k) t=pronadji(p->levi,k);
        else t=pronadji(p->desni,k);
    }
    return t;
}
```

```
struct drvo* naslednik(struct drvo *x){
    struct drvo *y;
    if(x->desni) {
        y=x->desni;
        while (y->levi) y=y->levi;
    }
    return y;
}
```

```

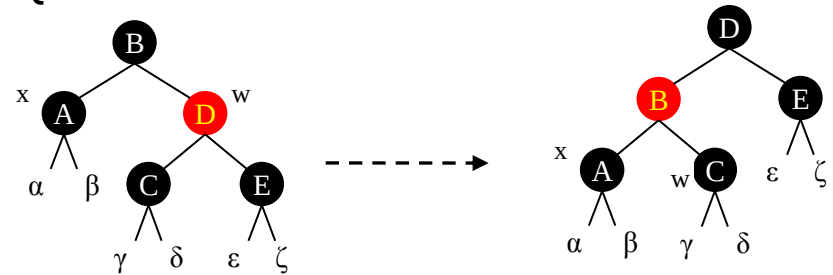
void obrisi_cc(struct drvo **p, struct drvo *z)
{
    struct drvo *x, *y;
    if ((!z->levi) || (!z->desni))
        y=z;
    else
        y=naslednik(z);
    if (y->levi)
        x=y->levi;
    else
        x=y->desni;
    if (x)
        x->roditelj=y->roditelj;
    if (!y->roditelj)
        *p=x;
    else
    {
        if (y==y->roditelj->levi)
            y->roditelj->levi=x;
        else
            y->roditelj->desni=x;
    }
    if (y!=z)
        z->broj=y->broj;
    if ((x) && (y->boja==CRNA))
    {
        BojaDelFix(p, x);
    }
    else if (y->boja==CRNA)
    {
        if (y->roditelj == *p)
        {
            struct drvo **pom;
            pom=p;
            if (y->roditelj->desni)
            {
                lrotacija(pom);
                if ((*pom)->desni && (*pom)->boja==CRNA)
                    BojaDelFix(p, (*pom)->desni);
            }
            else
            {
                drotacija(pom);
                if ((*pom)->levi && (*pom)->boja==CRNA)
                    BojaDelFix(p, (*pom)->levi);
            }
        }
        else
        {
            if (y->roditelj->desni)
                lrotacija(&(y->roditelj));
            else if (y->roditelj->levi)
                drotacija(&(y->roditelj));
        }
    }
}

```

```

void BojaDelFix(struct drvo **p, struct drvo *x){
    struct drvo *w,*t;
    while ((x!=(*p)) && (x->boja==CRNA)) {
        if (x==x->roditelj->levi) {
            w=x->roditelj->desni;
            if ((w) && (w->boja==CRVENA)) {
                w->boja=CRNA;
                x->roditelj->boja=CRVENA;
                t=x->roditelj;
                lrotacija(&t);
                if(!t->roditelj) *p=t;
                w=x->roditelj->desni;
            }
        }
    }

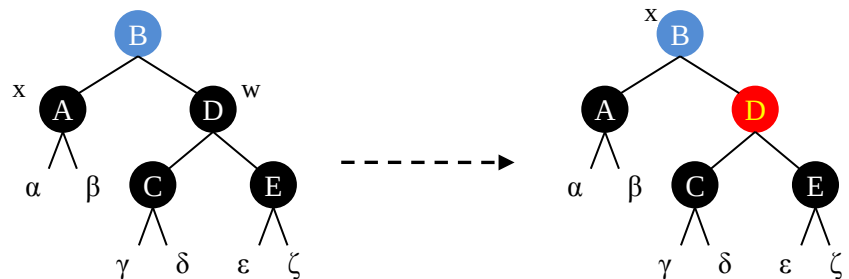
```



```

        if ((w->levi) && (w->desni)) {
            if((w->levi->boja==CRNA) && (w->desni->boja==CRNA)){
                w->boja=CRVENA;
                x=x->roditelj;
            }
        }
    }

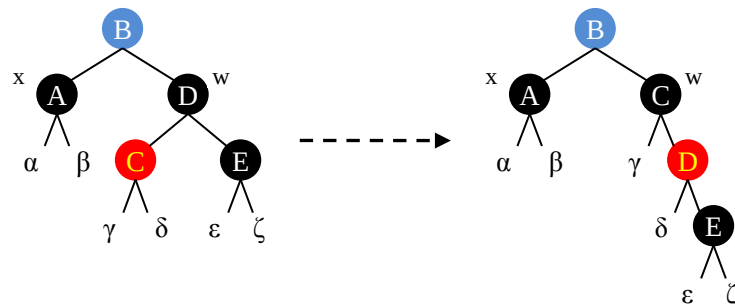
```



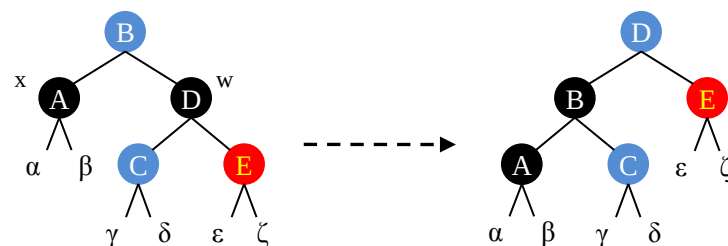
...

...

```
else { // if((w->levi->boja==CRNA) && (w->desni->boja==CRNA))
    if (w->desni->boja==CRNA) {
        w->levi->boja=CRNA;
        w->boja=CRVENA;
        drotacija(&w);
        if(!w->roditelj) *p=w;
        w=x->roditelj->desni;
```



```
    }
    w->boja=x->roditelj->boja;
    x->roditelj->boja=CRNA;
    w->desni->boja=CRNA;
    t=x->roditelj;
    drotacija(&t);
    if(!t->roditelj) *p=t;
    x=*p;
```



```
    }
```

```
}
```

```
}
```

...

...

```
else{                                     // if (x==x->roditelj->levi)
    w=x->roditelj->levi;
    if ((w) && (w->boja==CRVENA)) {
        w->boja=CRNA;
        x->roditelj->boja=CRVENA;
        t=x->roditelj;
        drotacija(&t);
        if(!t->roditelj) *p=t;
        w=x->roditelj->levi;
    }
    if ((w->levi) && (w->desni)) {
        if((w->levi->boja==CRNA) && (w->desni->boja==CRNA)){
            w->boja=CRVENA;
            x=x->roditelj;
        }
    }
}
```

...

```

else {
    // if((w->levi->boja==CRNA) && (w->desni->boja==CRNA))
    if (w->levi->boja==CRNA) {
        w->desni->boja=CRNA;  w->boja=CRVENA;
        lrotacija(&w);
        if(!w->roditelj) *p=w;
        w=x->roditelj->levi;
    }
    w->boja=x->roditelj->boja;
    x->roditelj->boja=CRNA;  w->levi->boja=CRNA;
    t=x->roditelj;
    lrotacija(&t);
    if(!t->roditelj) *p=t;
    x=*p;
}
}
}
}
x->boja=CRNA;
}

```