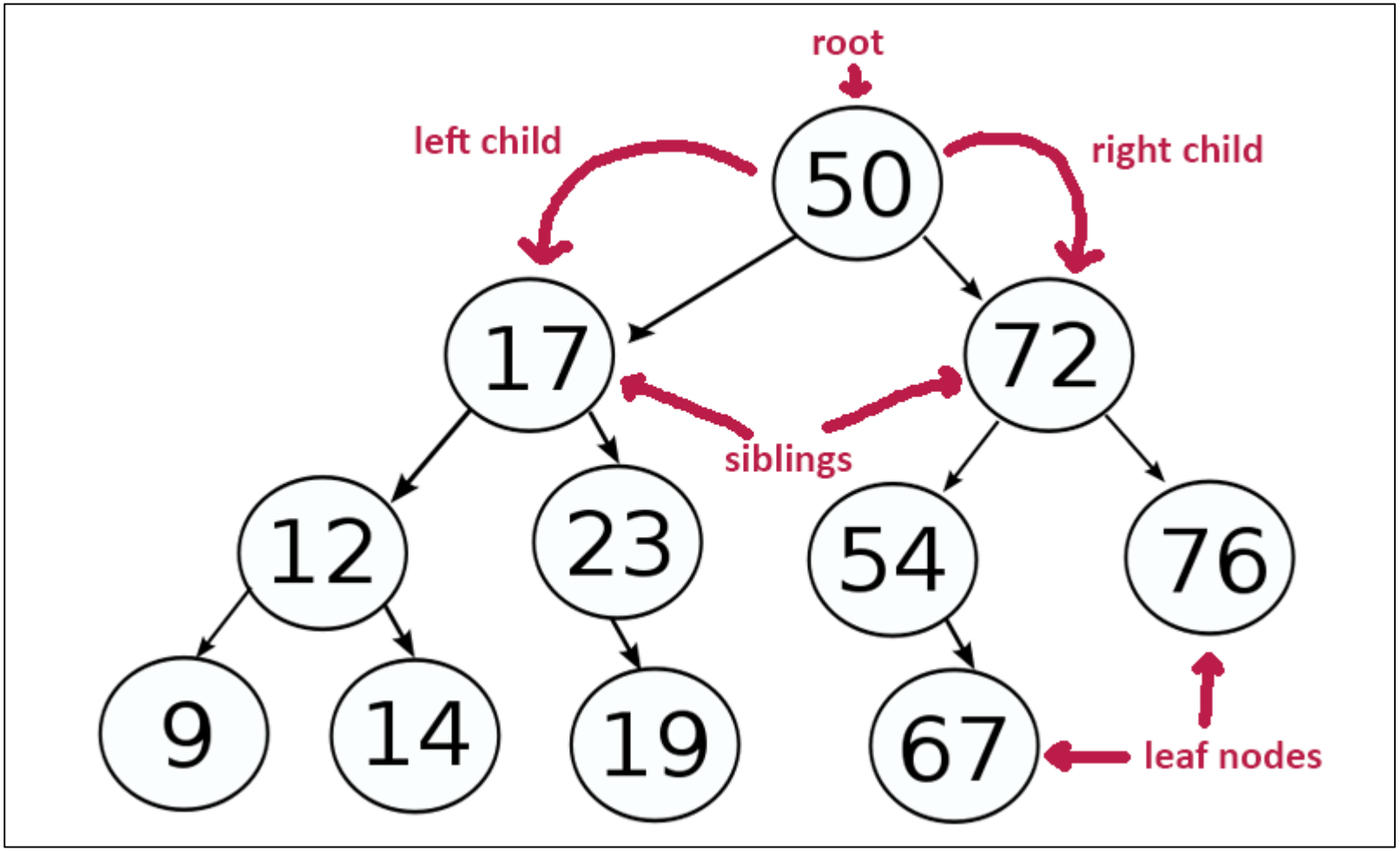
S08 – SD

Alin Zamfiroiu
alin.zamfiroiu@csie.ase.ro

Arbori binari de căutare

- Structura tara si structura nod
- Creare nod
- Inserare
- Parcurgere si afisare
- Numarul de nivele
- Afisare de pe un nivel
- Cautare nod
- Stergere

Arbori binari de căutare



http://www.bluesharktutorials.com/

Structura Tara si nod

```
struct tara{
    int id;
    char* nume;
    int nr_locuitori;
    float suprafata;
};
```

```
struct nod{
    tara info;
    nod* st;
    nod* dr;
};
```

- Pentru crearea unui nod se primesc cele trei informatii necesare: informatia, nodul din stanga si nodul din dreapta.

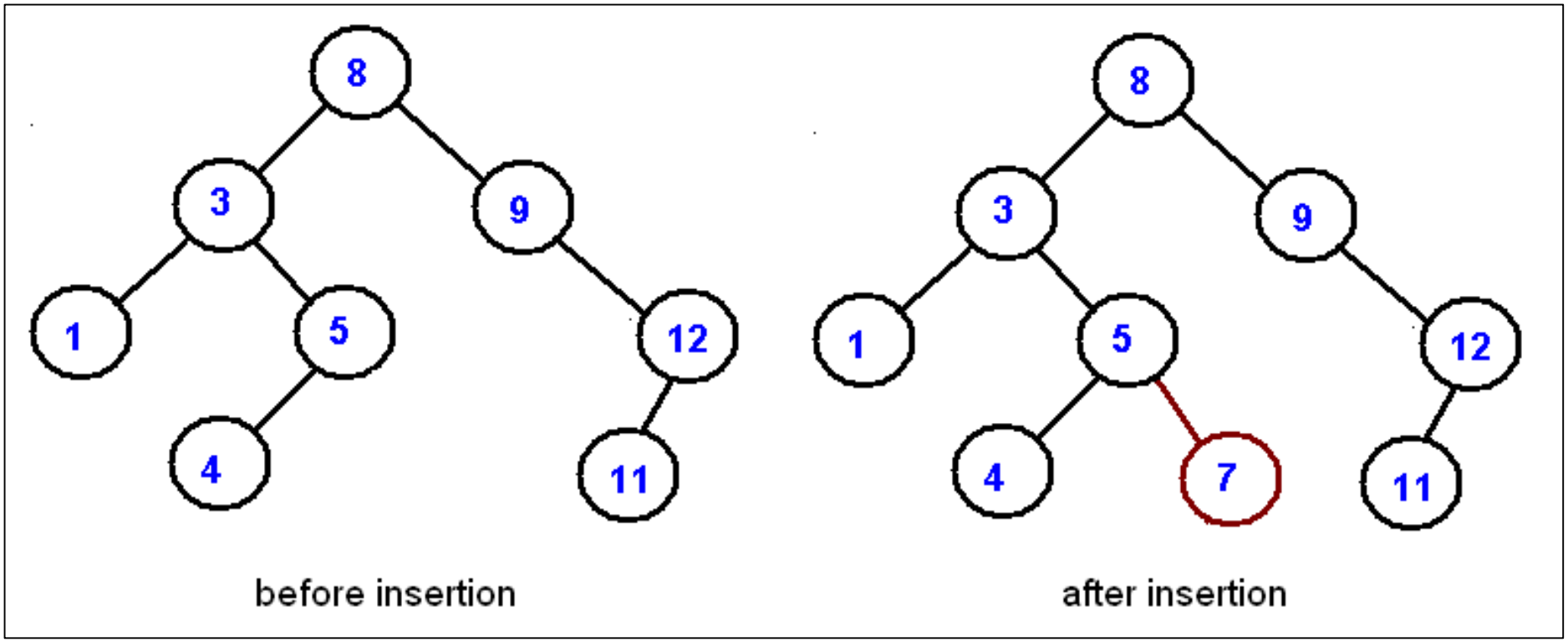
Creare nod

```
nod* creareNod(tara info, nod* st,nod*dr){
        nod* temp=(nod*)malloc(sizeof(nod));
        temp->info.id=info.id;
        temp->info.nume=(char*)malloc((strlen(info.nume)+1)*sizeof(char));
        strcpy(temp->info.nume,info.nume);
        temp->info.nr_locuitori=info.nr_locuitori;
        temp->info.suprafata=info.suprafata;

        temp->dr=dr;
        temp->st=st;

        return temp;
}
```

Inserare



http://www.cs.cmu.edu/

Inserare

```
nod* inserare(nod*radacina, tara info){
    if(radacina==NULL)
    {
        radacina=creareNod(info,NULL,NULL);
        return radacina;
    }
    else
    {
        if(info.id < radacina->info.id)
        {
            radacina->st=inserare(radacina->st,info);
        }
        else
        {
            radacina->dr=inserare(radacina->dr,info);
        }
        return radacina;
    }
}
```

Parcurgere si afisare

- Preordine ➔ **R**SD;
- Inordine ➔ S**R**D;
- Postordine ➔SD**R**.

Parcurgere si afisare

```
void SRD(nod*rad)
{
    if(rad)
    {
        SRD(rad->st);
        printf("%d.%s are %d locuitori si o supr
        SRD(rad->dr);
    }
}
```

Numarul de nivele

- Pentru determinarea numarului de nivele se calculeaza numarul de nivele pentru fiecare subarbore si se detemrina maximul dintre acestea, la care se adauga valoarea 1.

```
int max(int a,int b)
{
    if(a>b) return a;
    else
        return b;
}
```

Numarul de nivele

```
int nivele(nod*rad)
{
    if(rad)
        return 1+max(nivele(rad->st),nivele(rad->dr));
    else
        return 0;
}
```

Afisarea de pe un nivel

- Se furnizeaza ca si parametrii: radacina arborelui, nivelul de pe care dorim sa afisam informatiile si nivelul aferent radacinii (nivel 1).
- La fiecare apel atat pentru subarborele stang cat si pentru subarborele drept nivelul curent creste cu o unitate.
- Atunci cand cele doua nivele sunt egale, se afiseaza informatiile din nodul respectiv.

Afisarea de pe un nivel

```
void afisareNivel(nod*rad,int nivel,int nivelCurent)
{
    if(rad)
    {
        if(nivel==nivelCurent)
        {
            printf("%d.%s are %d locuitori si o suprafat
        }
        else
        {
            afisareNivel(rad->st,nivel,nivelCurent+1);
            afisareNivel(rad->dr,nivel,nivelCurent+1);
        }
    }
}
```

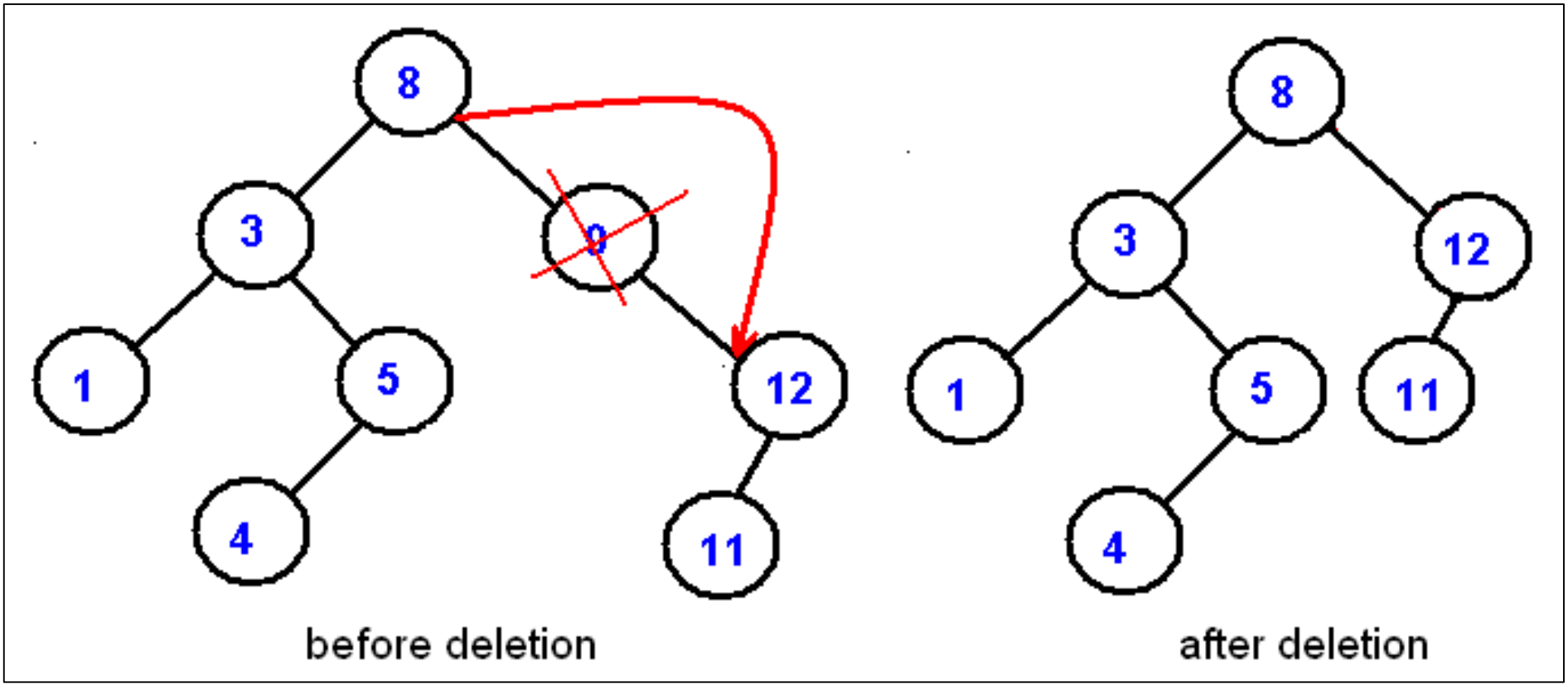
Cautare nod

- Pentru cautare se furnizeaza radacina arborelui si codul tarii de cautat sau tara ce este cautata.
- Cautarea se face dupa id-ul tarii deoarece arborele este sortat dupa acest id.

Cautare nod

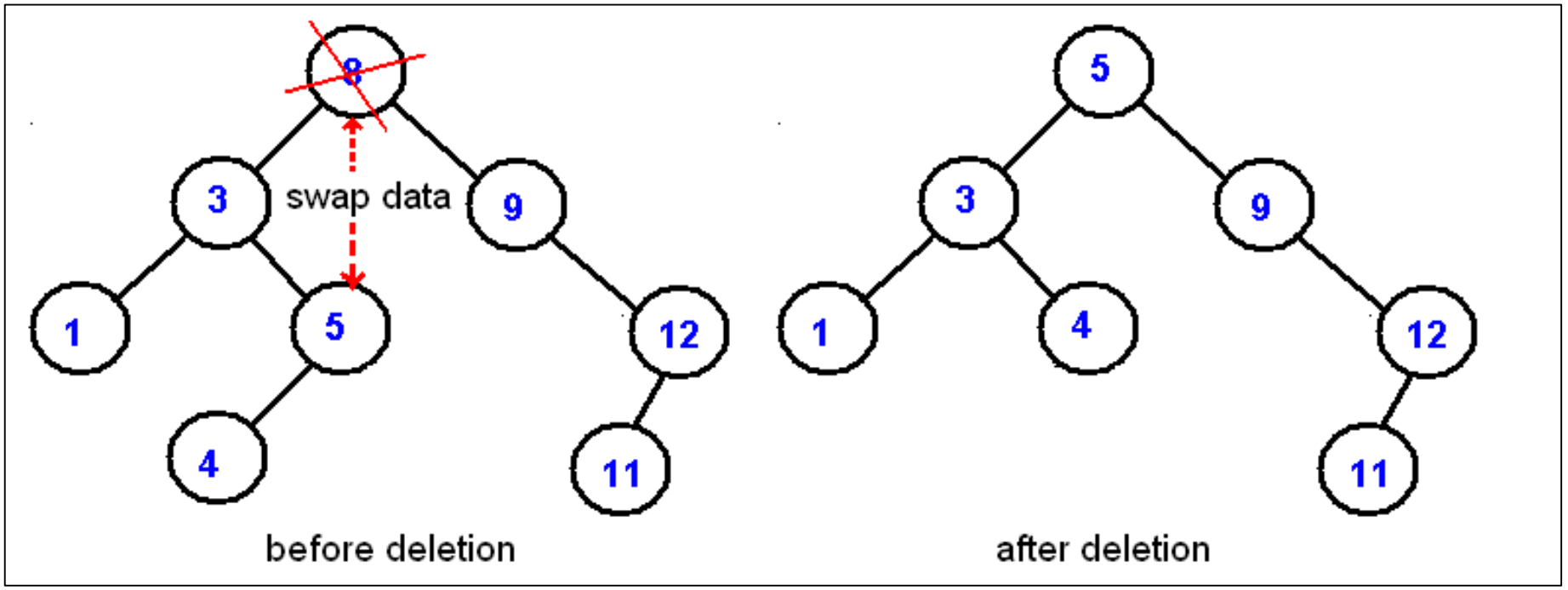
```
nod* cauta(nod*rad, int cod)
{
    if(rad)
    {
        if(rad->info.id==cod)
        {
            return rad;
        }
        else
        {
            if(rad->info.id>cod)
                return cauta(rad->st,cod);
            else
                return cauta(rad->dr,cod);
        }
    }
}
```

Stergere nod



http://www.cs.cmu.edu/

Stergere nod



http://www.cs.cmu.edu/