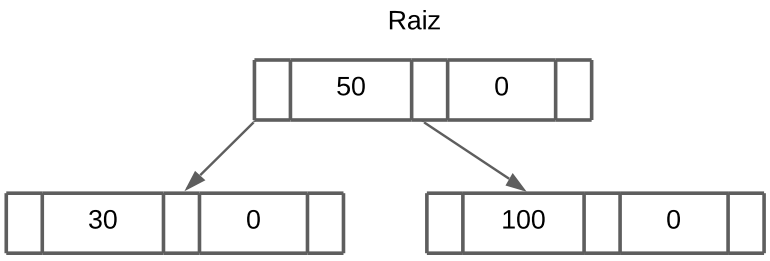
Então vamos ver como a função **adicionaNo()** funciona: Está função sera utilizada quando temos que inserir uma nova informação em um No que ja existe e esse No possui espaço.

Essa função recebe como parametro o No que queremos inserir a informação, a informação que sera inserida, e num No filho que pode ser NULL ou Não e a função devolverar o No com a informação inserida.

```
File  Edit  Selection  Find  View  Goto  Tools  Project  Preferences  Help
arvore23.c — Downloads
31  struct Arv23 *adicionaNo(struct Arv23 *No, int Info, struct Arv23 *Filho){
32
33    if (Info > (*No).Info1){
34      (*No).Info2 = Info;
35      (*No).dir = Filho;
36
37    }else{
38      (*No).Info2 = (*No).Info1;
39      (*No).dir = (*No).cen;
40      (*No).Info1 = Info;
41      (*No).cen = Filho;
42    }
43
44    (*No).NInfos = 2;
45
46  return No;
47  }
```

Agora vamos analizar a arvore abaixo e ver como a função se comporta, quando queremos **inserir o valor 25.**

info = 25

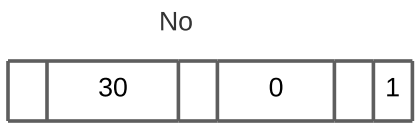


Vamos inserir o 25, entao percorreremos a arvore ate achar um No folha, o 25 sera inserido no No do valor 30, e como possui espaço chamaremos a função **adicionaNo()** para inserirmos o 25.

Então a função recebera como parametro o No do valor 30, a info que queremos inserir 25 e o filho que nesse caso será NULL.

info = 25

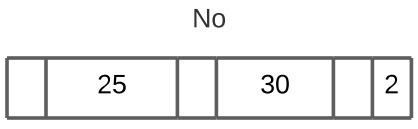
filho = NULL



Então será verificado se a info=25 que queremos inserir é maior que a info1=30 do No nesse caso não é então entra no else.

Na arvore 2-3 o valor da info1 tem que ser menor que o da Info2, então para que a arvore continue certa temos que fazer as seguintes operações:

- A info2 do No recebe a info1;
- Agora como a info1 era o maior valor, entao seus filhos do centro possuem valores maiores que a info1 e como agora a Info1 é a info2 temos que fazer com que o filho da direita receba o filho do centro do No.
- A info1 do No recebe a Info que queremos inserir;
- o filho do centro do No recebe o filho;
- E agora o numero de informações do No passa a ser 2;



E no fim é retornado o No pela função.