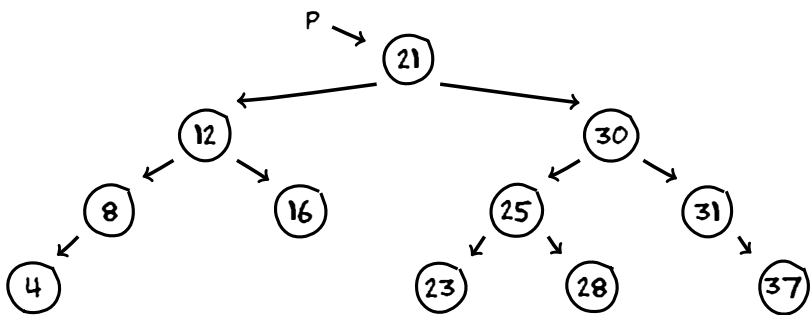


Imprimindo o k-ésimo menor

Imagine que o ponteiro **p** aponta para uma árvore binária de busca



A tarefa é

→ imprimir o **k-ésimo** menor elemento da árvore

No exemplo acima, para **k = 7**, isso nos daria

=> 25

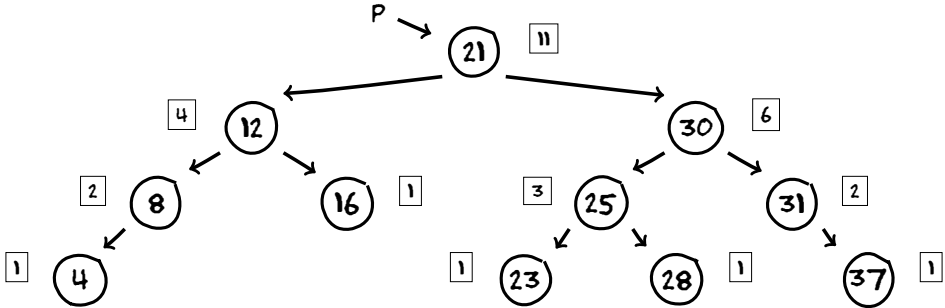
Bom, existe uma maneira simples de resolver essa tarefa.

Quer dizer, a gente pode percorrer a árvore contando os elementos por onde a gente passa.

Daí, quando o contador chega em **k**, a gente imprime o elemento.

Mas, a gente pode ser mais esperto do que isso — *(porque isso leva tempo $O(n)$)*

Para isso, a gente imagina que cada elemento armazena o número de elementos da sua subárvore no campo **.tam**.



Daí, se a gente quer imprimir o 7º elemento, então a gente não vai para o lado esquerdo

— *(porque lá só tem 4 elemntos)*

Então, a gente vai para o lado direito.

Mas quando a gente vai para o lado direito, a gente não pode esquecer quantos elementos ficaram para trás.

Nesse caso, ficaram 5 elementos para trás — *(4 na subárvore esquerda e mais a raiz)*

Daí, somando 5 com os 3 elementos da subárvore esquerda do 30, a gente obtém 8.

E assim, a gente descobre que o 7º elemento está lá.

Então agora a gente desce para a esquerda.

E nota que dessa vez não ficou nenhum elemento para trás — *(porque?)*

Daí, somando 5 com o único elemento na subárvore esquerda de 25, a gente obtém 6.

Isso significa que o 7º elemento não está lá.

Mas nós já contamos 6 elementos menores do que o elemento atual.

Então, o elemento atual é o 7º menor elemento da árvore.

E a gente pode imprimir ele

=> 25

Em resumo, a ideia é

1. descer pela árvore na direção do **k-ésimo** menor,
contando quantos elementos menores do que o atual já ficaram para trás
2. e imprimir o elemento atual quando o contador chega em **k**

A coisa fica assim

```
func IKE-Arv (p, k)                                     // Imprime k-ésimo Elemento

    q = p;    cont = 0

    Enquanto ( q != Nulo )                               // descida na árvore
        aux = 0
        Se (q->esq != Nulo)    aux = (q->esq)->tam        // filho esq.  no lugar do elemento
        Se (cont + aux ≥ k)    q = q->esq
        Sse (cont + aux == k-1)
            Imprime (q->num);  Retorna
        Senão
            cont = cont + aux + 1;    q = q->dir
```