

O Programa 6.8 apresenta o refinamento final do procedimento *Inser*.

Programa 6.8 Refinamento final do algoritmo *Inser*

```

procedure Inser (Reg: TipoRegistro; var Ap: TipoApontador);
var Cresceu: Boolean; RegRetorno: TipoRegistro;
    ApRetorno, ApTemp: TipoApontador;

procedure Ins (Reg: TipoRegistro; Ap: TipoApontador; var Cresceu: Boolean;
    var RegRetorno: TipoRegistro; var ApRetorno: TipoApontador);
var i, j: Integer; ApTemp: TipoApontador;
begin
    if Ap = nil
    then begin Cresceu := true; RegRetorno := Reg; ApRetorno := nil; end
    else with Ap do
        begin
            i := 1;
            while (i < n) and (Reg.Chave > r[i].Chave) do i := i + 1;
            if Reg.Chave = r[i].Chave
            then begin
                writeln (' Erro: Registro ja esta presente'); Cresceu := false;
            end
            else begin
                if Reg.Chave < r[i].Chave then i := i - 1;
                Ins (Reg, p[i], Cresceu, RegRetorno, ApRetorno);
                if Cresceu
                then if n < mm
                then begin { Pagina tem espaco }
                    InserNaPagina (Ap, RegRetorno, ApRetorno);
                    Cresceu := false;
                end
                else begin { Overflow: Pagina tem que ser dividida }
                    new (ApTemp);
                    ApTemp.n := 0; ApTemp.p[0] := nil;
                    if i < M + 1
                    then begin
                        InserNaPagina (ApTemp, r[mm], p[mm]);
                        n := n - 1;
                        InserNaPagina (Ap, RegRetorno, ApRetorno)
                    end
                    else InserNaPagina (ApTemp, RegRetorno, ApRetorno);
                    for j := M + 2 to mm do
                        InserNaPagina (ApTemp, r[j], p[j]);
                    n := M; ApTemp.p[0] := p[M + 1];
                    RegRetorno := r[M + 1]; ApRetorno := ApTemp;
                end;
            end;
        end;
    end;
end; { Ins }

```

Continuação do Programa 6.8

```

begin
    Ins (Reg, Ap, Cresceu, RegRetorno, ApRetorno);
    if Cresceu
    then begin { Arvore cresce na altura pela raiz }
        new (ApTemp);
        ApTemp.n := 1;
        ApTemp.r[1] := RegRetorno;
        ApTemp.p[1] := ApRetorno;
        ApTemp.p[0] := Ap; Ap := ApTemp
    end
end; { Inser }

```

A Figura 6.11 mostra o resultado obtido quando se insere uma sequência de chaves em uma árvore B de ordem 2: a árvore da Figura 6.11(a) é obtida após a inserção da chave 20, a árvore (b) é obtida após a inserção das chaves 10, 40, 50 e 30 na árvore (a), a árvore (c) é obtida após a inserção das chaves 55, 3, 11, 4, 28, 36, 33, 52, 17, 25 e 13 na árvore (b) e, finalmente, a árvore da parte (d) é obtida após a inserção das chaves 45, 9, 43, 8 e 48.

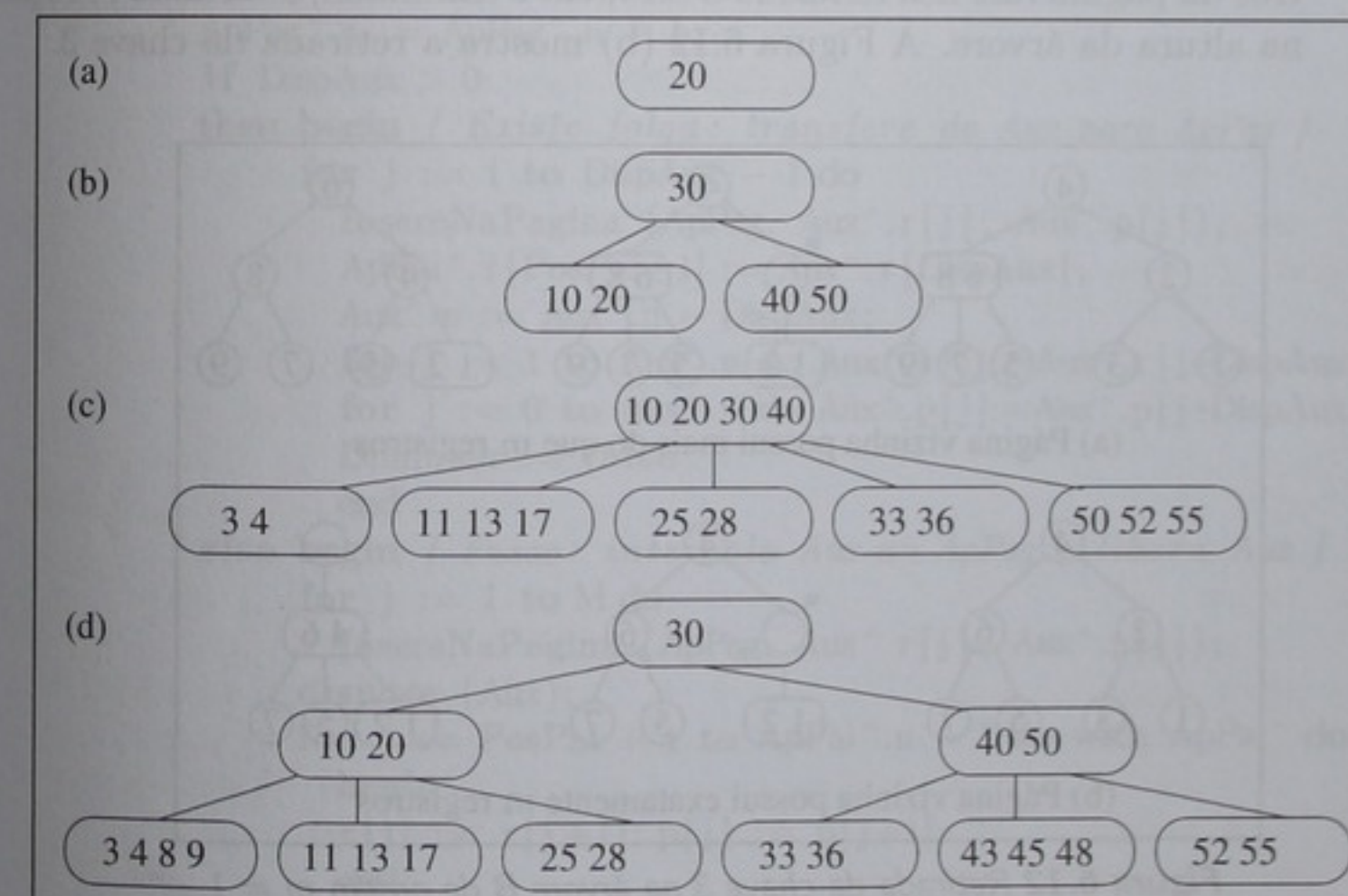


Figura 6.11 Crescimento de uma árvore B de ordem 2.

A última operação a ser estudada é de **retirada**. Quando a página que contém o registro a ser retirado é uma página folha, a operação é simples. No caso de não ser uma página folha, o registro a ser retirado deve ser primeiro substituído por um registro contendo uma chave adjacente (antecessora ou sucessora), como