

Divide y Vencerás

Estructuras de Datos II

Erwin Meza Vega

Divide y vencerás

- Obtener una solución de un problema a partir de la solución de subproblemas del mismo tipo.
 - Si los subproblemas son aún muy grandes, se subdividen nuevamente
 - Si los subproblemas son suficientemente sencillos, se pueden solucionar directamente
- Generalmente utilizan recursividad

Proceso general

- División: Plantear el problema de tal forma que pueda ser descompuesto en k subproblemas del mismo tipo
 - Necesariamente $1 \leq k \leq n$
- Resolver los subproblemas de forma independiente
 - Directamente
 - Utilizando recursividad
- Combinar las soluciones obtenidas para construir la solución al problema original

Algoritmo general

```
PROCEDURE DyV(x:TipoProblema):TipoSolucion;  
  VAR i,k,:CARDINAL;  
  s:TipoSolucion;  
  subproblemas: ARRAY OF TipoProblema;  
  subsoluciones:ARRAY OF TipoSolucion;  
  BEGIN  
    IF EsCasobase(x) THEN  
      s:=ResuelveCasoBase(x)  
    ELSE  
      k:=Divide(x,subproblemas);  
      FOR i:=1 TO k DO  
        subsoluciones[i]:=DyV(subproblemas[i])  
      END;  
      s:=Combina(subsoluciones)  
    END;  
    RETURN s  
  END DyV;
```

Algoritmo general

- k representa el número de subproblemas
 - Debe ser pequeño e independiente de la entrada
 - Si se tiene solo una llamada recursiva ($k=1$): Algoritmos de simplificación (Factorial, búsqueda binaria, etc)
- Diseño simple, claro y robusto
- Llevan mayor tiempo de ejecución que los algoritmos iterativos

Búsqueda binaria no centrada

```
PROCEDURE BuscBin2(Var a:vector;  
  prim,ult:CARDINAL;x:INTEGER):BOOLEAN;  
  VAR tercio:CARDINAL; (* posicion del elemento n/3 *)  
  BEGIN  
    IF (prim>=ult) THEN RETURN a[ult]=x  
    ELSE  
      tercio:=prim+((ult-prim+1)DIV 3);  
      IF x=a[tercio] THEN RETURN TRUE  
      ELSIF (x<a[tercio]) THEN RETURN BuscBin2(a,prim,tercio,x)  
      ELSE RETURN BuscBin2(a,tercio+1,ult,x)  
    END  
  END  
END  
END BuscBin2;
```