

Vuelta atrás (backtracking)

Si sol es una solución terminamos

Construimos el conjunto de posibles continuaciones

Si es vacío, nuestra solución parcial no lleva a ninguna solución y devolvemos null

Si hay alguna posible continuación, la escogemos y continuamos de forma recursiva con ella.

Si la llamada recursiva nos devuelve una solución terminamos, sino continuamos con la siguiente posible continuación.

ESQUEMA VueltaAtrás

```
Solucion backtracking(Solucion sol){  
    if (esSolucion(sol)) return sol;  
    else{  
        Cont setCont = posContinuacion(sol);  
        Solucion solAux = null;  
        while (!esVacia(setCont) && solAux==null){  
            cont = selecciona(setCont);  
            setCont = SetCont - {cont};  
            solAux=backtracking(sol+cont);  
        }  
        return solAux;  
    }  
}
```

Backtracking por enumeración

ESQUEMA VueltaAtrás (encontrar el número de soluciones)

```
int backtracking(Solucion sol){  
    if (esSolucion(sol)) return 1;  
    else{  
        Cont setCont = posContinuacion(sol);  
        int num = 0;  
        while (!esVacia(setCont)){  
            cont = selecciona(setCont);  
            setCont = SetCont - {cont};  
            int num1=backtracking(sol+cont);  
            num += num1  
        }  
        return num;  
    }  
}
```

Backtracking por enumeración

ESQUEMA VueltaAtrás (encontrar todas las soluciones)

```
void backtracking(List<Solucion> listaSol, Solucion sol){  
    if (esSolucion(sol)) listaSol.add(sol);  
    else{  
        Cont setCont = posContinuacion(sol);  
        while (!esVacia(setCont)){  
            cont = selecciona(setCont);  
            setCont = SetCont - {cont};  
            backtracking(listaSol, sol+cont);  
        }  
    }  
}
```

Backtracking para optimización

ESQUEMA VueltaAtrás (la mejor solución)

```
Solucion backtracking(Solucion sol, Solucion msol, int mcal){  
    if (esSolucion(sol)) {  
        int cal = calidad(sol);  
        if (mcal > cal) return msol;  
        else return sol;  
    };  
    else{  
        Cont setCont = posContinuacion(sol);  
  
        while (!esVacia(setCont)){  
            cont = selecciona(setCont);  
            setCont = SetCont - {cont};  
            Solucion otra=backtracking(sol+cont,msol,mcal);  
            if (calidad(otra)>mcal) msol = otra; mcal = calidad(otra)  
        }  
        return msol;  
    }  
}
```