

```

function 3DBPP(items, W, H, D)
  bins  $\leftarrow$   $\emptyset$ ;
  items  $\leftarrow$  items  $\cup$  GENERATESUPERITEMS(items);
  repeat
    packedItems  $\leftarrow$  PACKBIN(items, W, H, D) ;
    bin  $\leftarrow$  COMPLETEBIN(packedItems, W, H, D);
    bins  $\leftarrow$  bins  $\cup$  bin;
    items  $\leftarrow$  items  $\setminus$  packedItems;
  until items =  $\emptyset$ ;
  return bins;
end function

```

```

function PACKBIN(items, W, H, D)
  packed  $\leftarrow$   $\emptyset$ ;
  currentPacking  $\leftarrow$   $\emptyset$ ;
  toPack  $\leftarrow$  items;
  planes  $\leftarrow$   $\{(0, \emptyset, \emptyset)\}$ ;  $\triangleright$  PriorityQueue, (z, supportItems, upperItems)
  repeat
    p  $\leftarrow$  DEQUEUE(planes);  $\triangleright$  Polls the lowest plane from the set
    currentPacking  $\leftarrow$  PACKPLANE(p, toPack, W, H, D);
    packed  $\leftarrow$  packed  $\cup$  currentPacking;
    toPack  $\leftarrow$  toPack  $\setminus$  packed;
    UPDATEPLANES(planes, currentPacking, 5);
  until planes =  $\emptyset$   $\vee$  toPack =  $\emptyset$   $\vee$  currentPacking =  $\emptyset$ ;
  return packed;
end function

```

```

procedure UPDATEPLANES(planes, packed, tolerance)
  for item in packed do
    if  $\nexists p \in \text{planes} : |p.z - \text{item}.z| \leq \text{tolerance}$  then
      planes  $\leftarrow$  planes  $\cup$  (item.z,  $\emptyset$ ,  $\emptyset$ );
    end if
    for p  $\in$  planes :  $0 \leq p.z - (\text{item}.z + \text{item}.w) \leq \text{tolerance}$  do
      p.supportItems  $\leftarrow$  p.supportItems  $\cup$  item;
    end for
    for p  $\in$  planes : p.z < (item.z + item.w) do
      p.upperItems  $\leftarrow$  p.upperItems  $\cup$  item;
    end for
  end for
end procedure

```

```

function PACKPLANE(plane, toPack, W, H, D)
  packed  $\leftarrow \emptyset$ ;
  return packed;
end function
function SCOREPACKING(packed, W, H, D)
  A  $\leftarrow$  SUMAREA(packed);
  V  $\leftarrow$  SUMVOLUME(packed);
end function

```