4. TVS + Job Scheduling

Sunday, February 21, 2021 8:32 AM

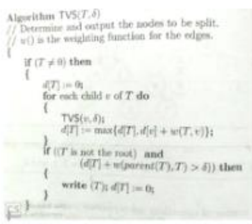
Greedy

1. Knapsack
2. Tree vertex splitting
3. Job scheduling with deadlines
4. Minimum spanning tree

Greedy

TVS

1. For every node assign a value
2. For node u, d(v) + weight(u, w) >= delta, then split the node
   w-> parent
   Value of leaf node is always zero
3. All values of d(v) initialized to zero (initially)
4. d(T) = max(d(T), d(v)+w(T,v)) [for non-leaf nodes]



```
Algorithm TVS(T, δ)
// Determine and output the nodes to be split.
// w() is the weighting function for the edges.
{
    if (T ≠ 0) then
    {
        d[T] := 0;
        for each child v of T do
        {
            TVS(v, δ);
            d[T] := max{d[T], d[v] + w(T, v)};
        }
        if ((T is not the root) and
                (d[T] + w(parent(T), T) > δ)) then
        {
            write (T); d[T] := 0;
        }
    }
}
```

Job scheduling with deadlines:

- Profit for each job
- Job number
- Slot of execution

Find which job fits into the right slot

Arrange in order of decreasing profit

Job number	Profit	Slot number
1	20	2
2	15	2
5	10	1
4	5	3
3	1	3

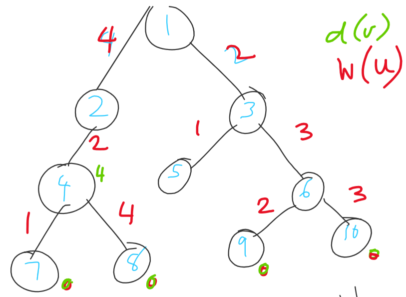
Job 1 can execute in slot 1,2 not 3
Job 3 can execute in slot 1,2,3

Maximal slot: 3
Draw the graph:

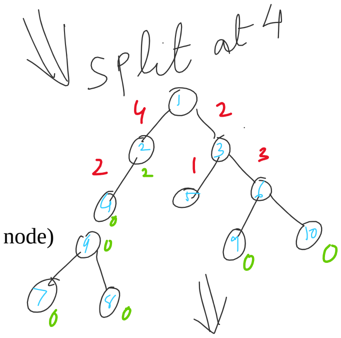
Job	Slot allotted
1	1
2	2
5	Reject
4	3
3	Reject

Job	Slot	Job considered	Profit	Slot assigned
[]	-	1	0	0-1
[1]	0-1	2	20	1-2
[1 2]	0-1, 1-2	5	35	Reject
[1 2]	0-1, 1-2	4	35	2-3
[1 2 4]	0-1, 1-2, 2-3	3	40	reject

Job number	Profit	Slot number
1	20	4
2	15	3
5	10	2
4	5	1



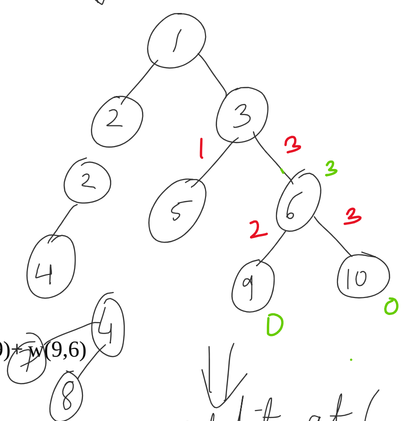
Max(w, child nodes)
d(4) + w(4,x)
0 + max (1, 4) = 4



If in node 4,
d(4) + 2 >= 5?
Yes split at 4
d(4) = 0 after split (leaf node)

d(7) + 1 >= 5?
No , no split at 7

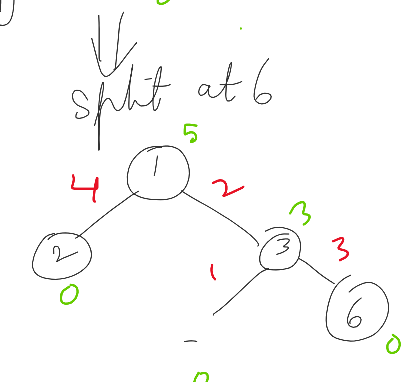
d(8) + 4 > = 5?
No, no split at 8



d(2) = d(2), d(4) + max(w(4,2))
= 0 + 0 + (max(2)) = 2
For node 2,
d(2) + w(2,1) >=5?
Yes, split at 2

For node 1, d(3) to be calculated
POST ORDER

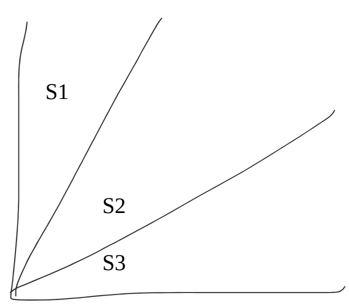
At node 6,
d(6) = d(6), max(d(10)+w(10,6),d(9)+w(9,6)
= 6
d(6) + w(6,3) = 3 + 3 >=5?
Split at 6

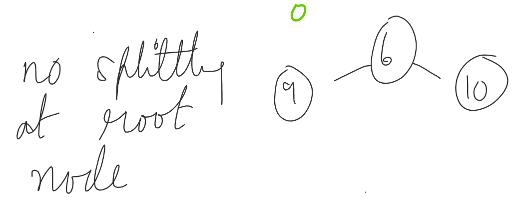


No splitting at root node

Complexity:

Using arrays : O(v^2)
Using linked/adjacency list (singly): O(v)





3	1	4

Job	Slot
1	1
2	2
3	2 (2 pushed to slot 3)
4	1 (1 pushed to 4)