Branch & Bound (BB) (Algorithm design technique)

State space tree

Blind search

1. FIFO BB (Queue)
2. LIFO BB (Stack)

3. Least Cost BB

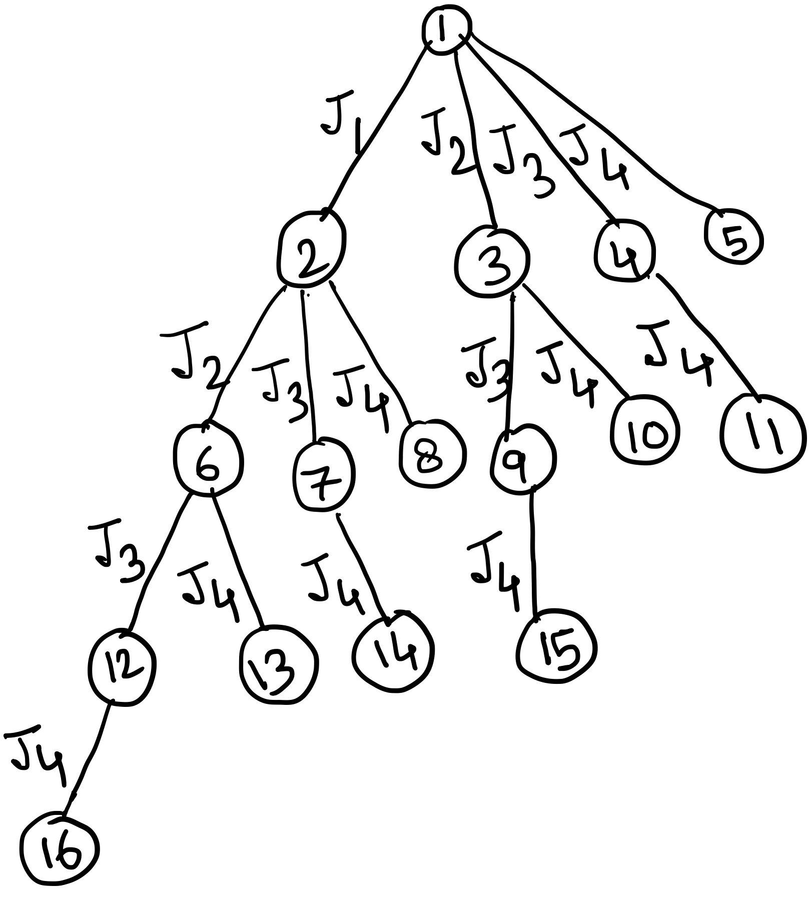
Job Sequencing with deadline

Jobs (J) = { J_1, J_2, J_3, J_4 }

Penalty (P) = {5, 10, 2, 7}

Deadline (D) = {2, 1, 1, 3}

1. FIFO BB

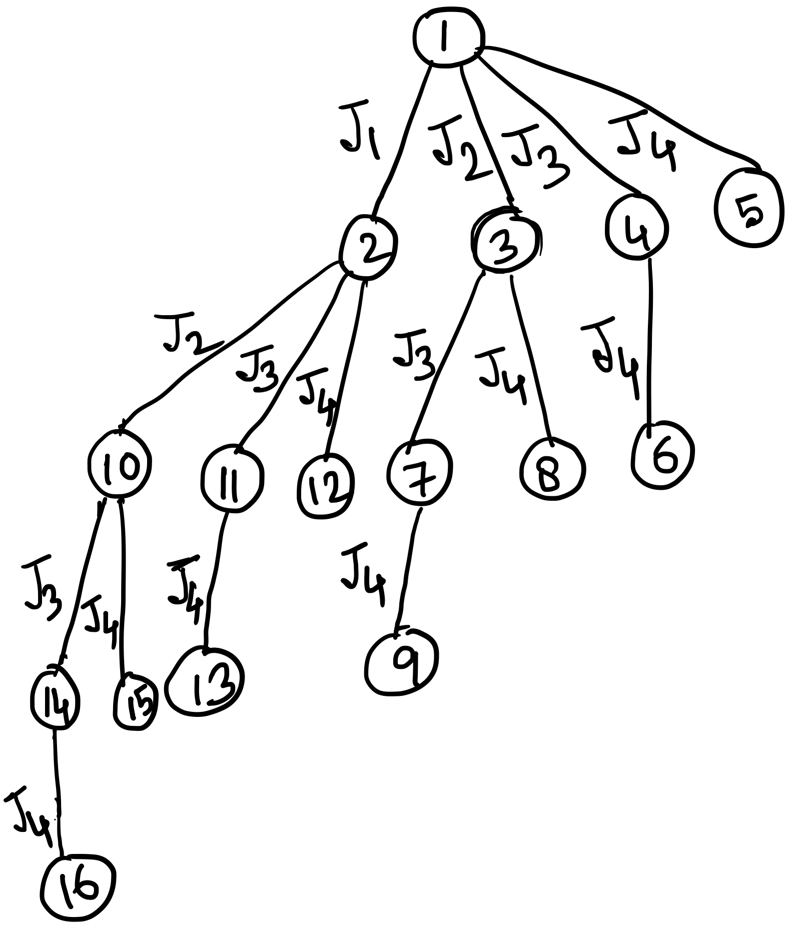


Queue

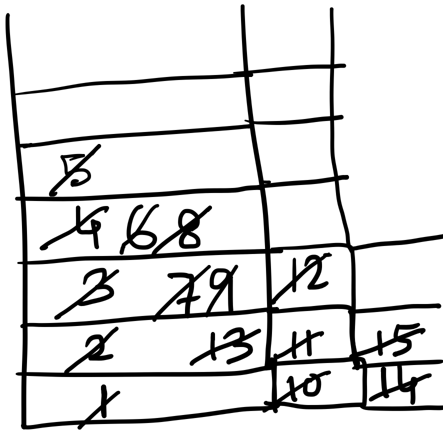
~~1~~	~~2~~	~~3~~	~~4~~	~~5~~	~~6~~	~~7~~	~~8~~	~~9~~	~~10~~	~~11~~

~~12~~	~~13~~	~~14~~	~~15~~	~~16~~

2. LIFO BB :



Stack



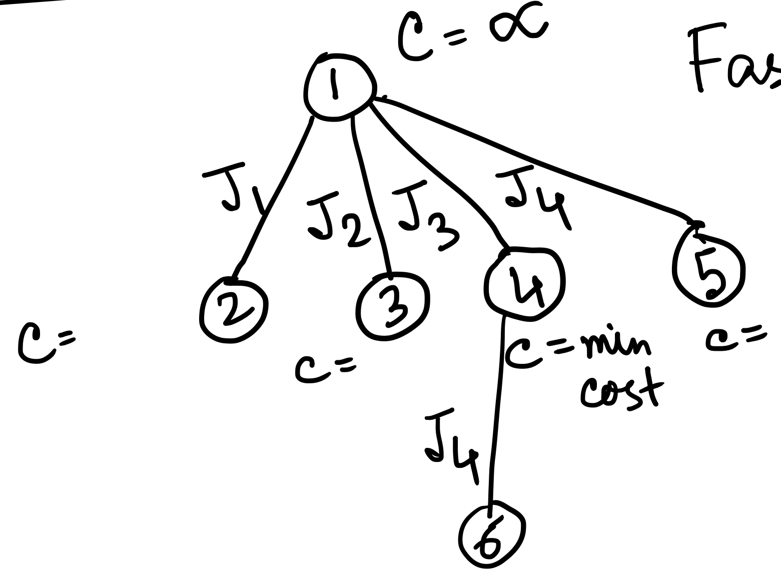
3. Least Cost BB :

Intelligent search

Fast search technique

↓

to find the solution very quickly.



```
                        ← Problem
Algorithm BB_FIFO (P, root, goal)
  Begin
      Q := Null
      enqueue(Q, root) →
      best := value at root
      while (Q!:= Null) then
          v := dequeue(Q)
          for all children u of v do
              if bound of u is better than the
                                  best then
                  enqueue(Q, u) →
                  update the best value
              end if
          end for
      end while
  End
```

```
→ S := Null   BB_LIFO
              PUSH S
              POP
```

```
Priority Queue
BB_LeastCost
v := Extract_Min(Q) ←
if bound of v is ←
better than the best
   then

end if
```

Job Sequencing with Deadlines

global upper bound → upper = ~~∞~~ ~~19~~ ~~14~~ ~~9~~ 8

Jobs = $\{J_1, J_2, J_3, J_4\}$

Penalties (P) = $\{5, 10, 6, 3\}$

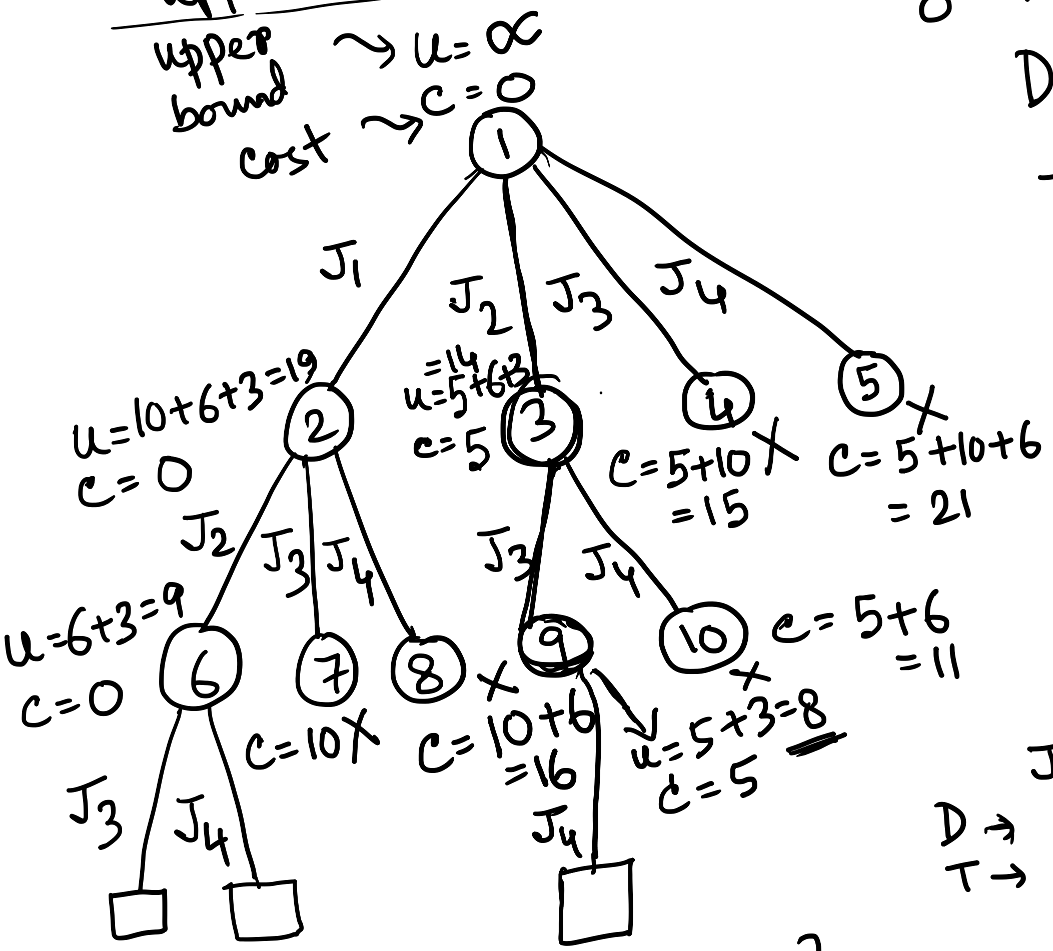
Deadlines (D) = $\{1, 3, 2, 1\}$

Time (T) = $\{1, 2, 1, 1\}$

Objective is to minimize the penalty

u = Sum of all penalties except that included in soln.

c = Sum of penalties till the last job considered



	J_1	J_2	J_3	J_1	J_2	J_4
D →	1	3	2	1	3	1
T →	1	2	1	1	2	1

Jobs done ⇒ $\{J_2, J_3\}$

Penalty to give ⇒ 8 units (J_1, J_4)

	J_2	J_3	J_4
D →	3	2	1
T →	2	1	1