Backtracking and Branch and Bound

Module 11

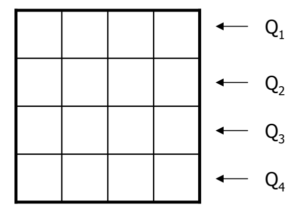
CSE5311 Fall 2008

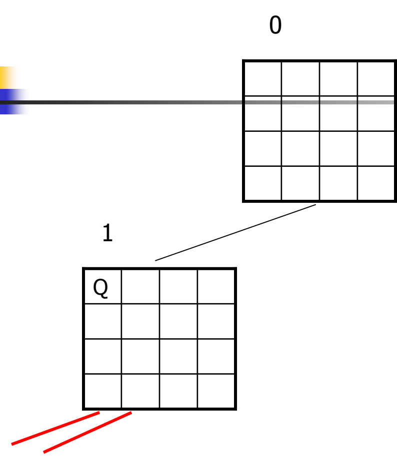
Backtracking

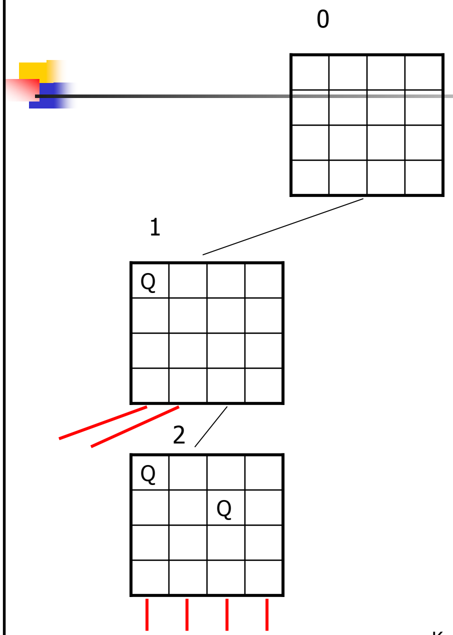
- Using Backtracking
  - *Large instances of difficult combinatorial problems can be solved*
  - *Worst case complexity of Backtracking can be exponential*
- Typically, a path is taken to check if a solution can be reached
  - *If not, the path is abandoned and another path taken*
  - *The process is repeated until the solution is arrived at*

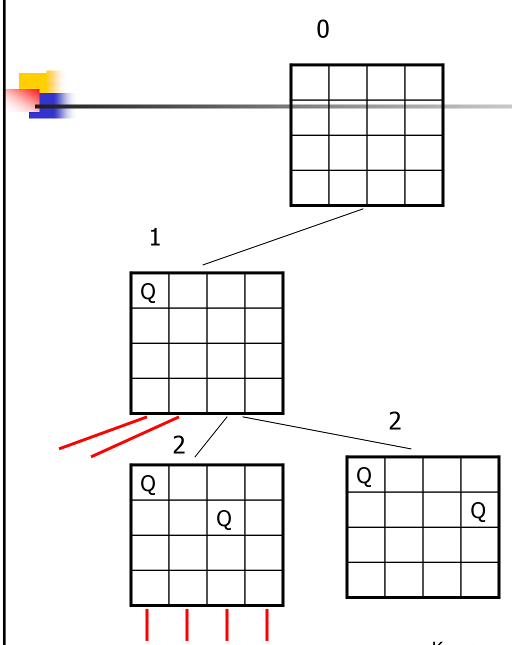
N-Queens problem

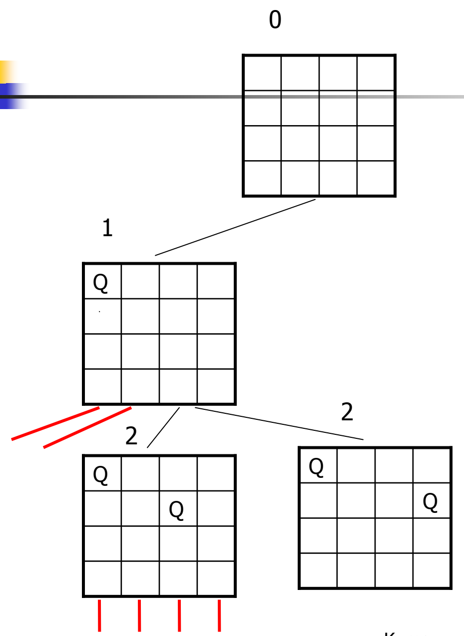
- Place n-queens on an n × n chess board so that no two queens attack each other.
  - *A queen can attack another if the latter is on the same row, column or diagonal*

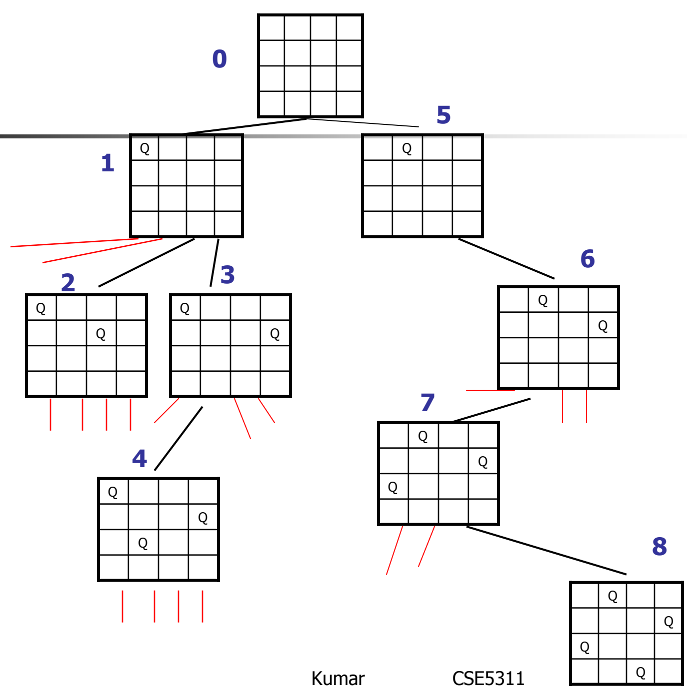




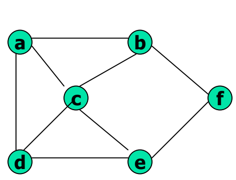

0
1
Q
2
Q
Q

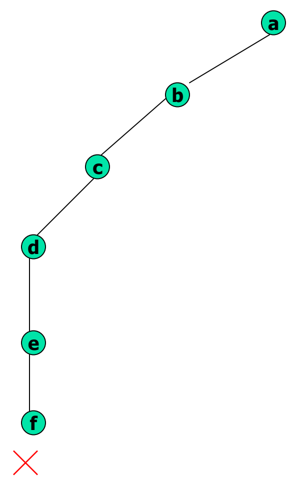

0
1
Q
2
Q
Q
2
Q
Q


0
1
Q
2
Q
Q
2
Q
Q


0
5
1
6
2
3
7
4
8

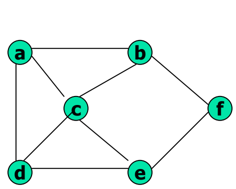
Hamiltonian Circuit Problem

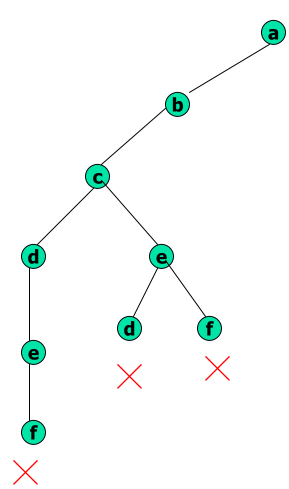




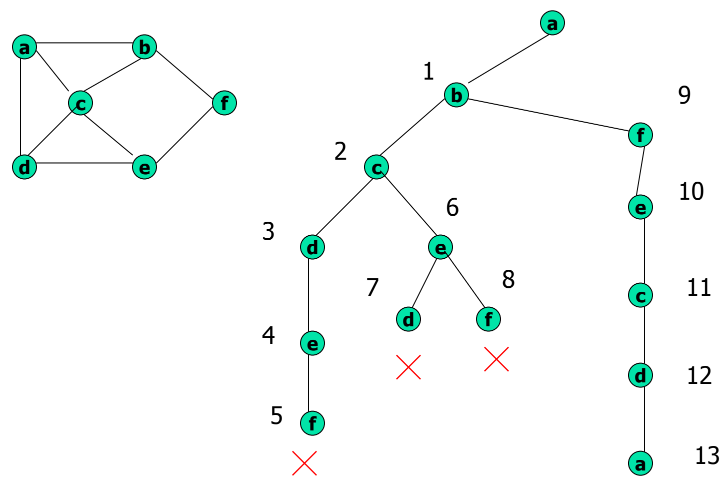
Start at a vertex and visit all the other vertices in the graph exactly once and return to the start vertex

Hamiltonian Circuit Problem





Hamiltonian Circuit Problem

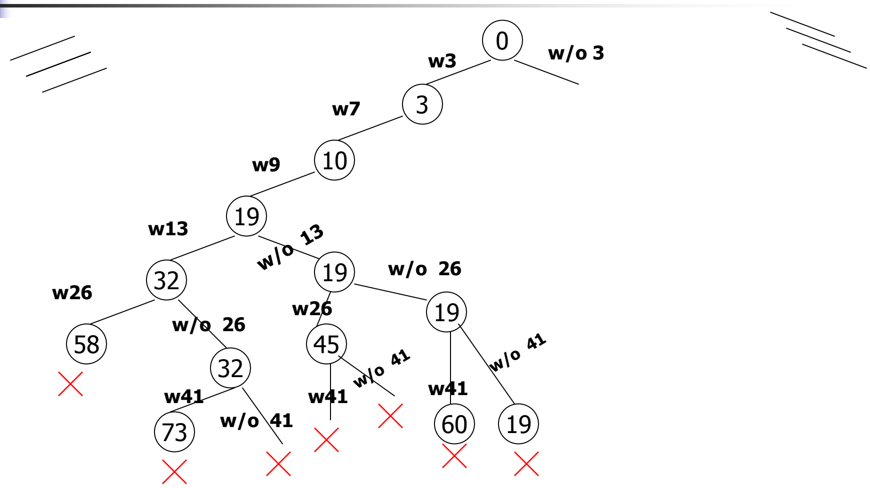


Subset Sum Problem

- Given a Set S ={s1,s2, … Sn} and a posiitive integer ‘d’ find a subset of the given set S such that the sum of the positive integers in the subset is equal to ‘d’.
- Let S = {3,7,9,13,26,41}; d = 51.
- Note – the list should be sorted.

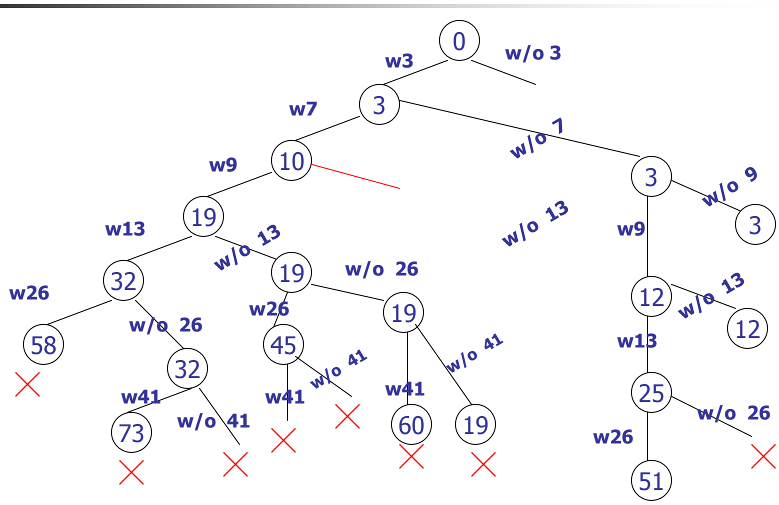
Subset problem

Let S = {3,7,9,13,26,41};
d = 51



Subset problem

Let S = {3,7,9,13,26,41};
d = 51



Branch and Bound

- With backtracking
  - *The search space is can be very large*
  - *It is an exhaustive search*
  - *Worst case complexity is exponential*
- Branch and bound technique
  - *Limits the search space*
  - *Through an estimate of the*
    - Upper bound or
    - Lower bound

Scheduling problem

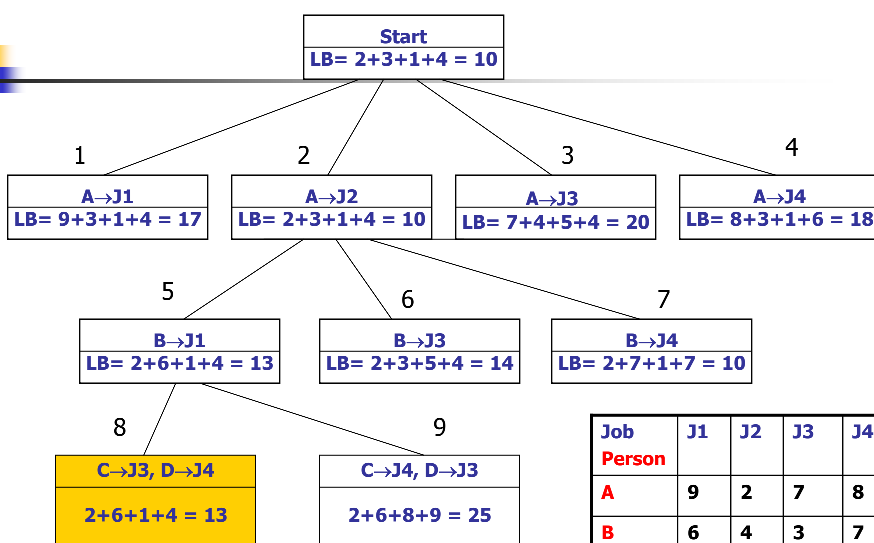
- The problem of assigning n people to n jobs such that the total cost is as small as possible

Job / Person	J1	J2	J3	J4
A	9	2	7	8
B	6	4	3	7
C	5	8	1	8
D	7	6	9	4

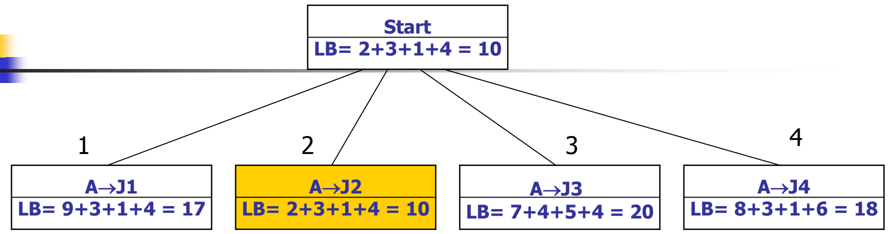
Branch and Bound

- Find a Lower Bound on the cost of the solution
- The lower bound is only an estimate
  - *This is only an estimate*
  - *The LB may not be a legitimate solution*
- In this case, consider the lowest cost from each row
  - *2 +3+1+4 =10*
  - *This is our LB*

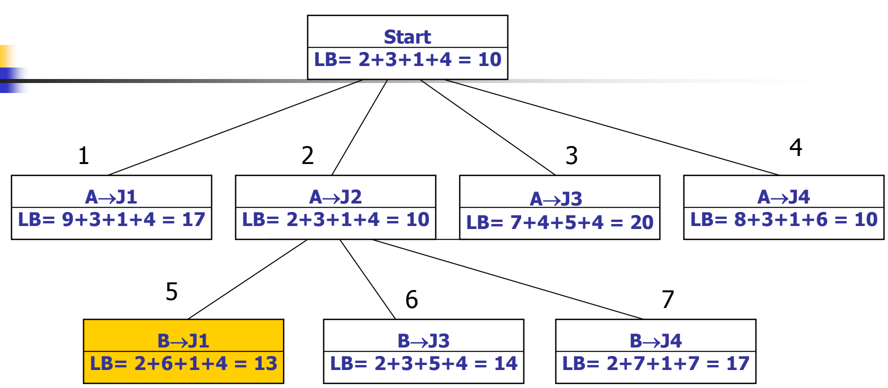
Job / Person	J1	J2	J3	J4
A	9	2	7	8
B	6	4	3	7
C	5	8	1	8
D	7	6	9	4



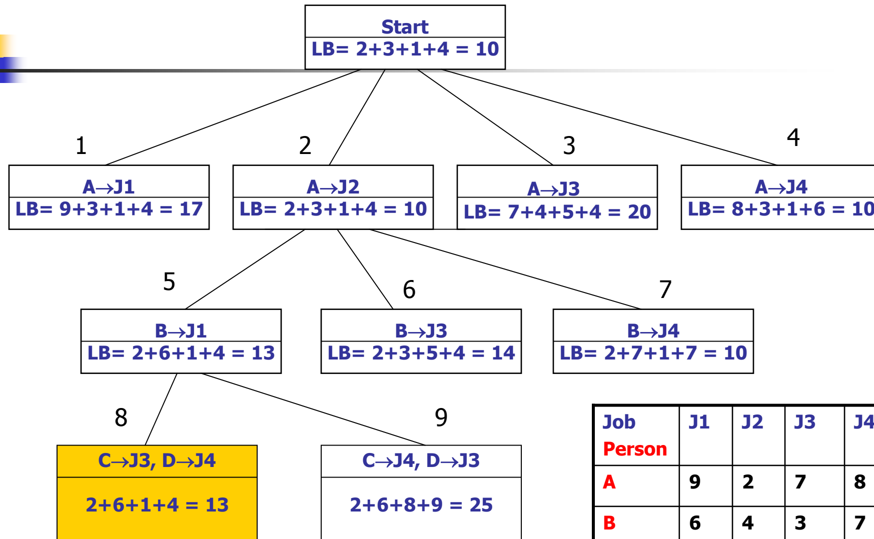
Job / Person	J1	J2	J3	J4
A	9	2	7	8
B	6	4	3	7
C	5	8	1	8
D	7	6	9	4



Job Person	J1	J2	J3	J4
A	9	2	7	8
B	6	4	3	7
C	5	8	1	8
D	7	6	9	4



Job Person	J1	J2	J3	J4
A	9	2	7	8
B	6	4	3	7
C	5	8	1	8
D	7	6	9	4



Job / Person	J1	J2	J3	J4
A	9	2	7	8
B	6	4	3	7
C	5	8	1	8
D	7	6	9	4

Knapsack Problem

- We wish the maximize the profit in the knapsack
- Maximization
- Use Upper bound

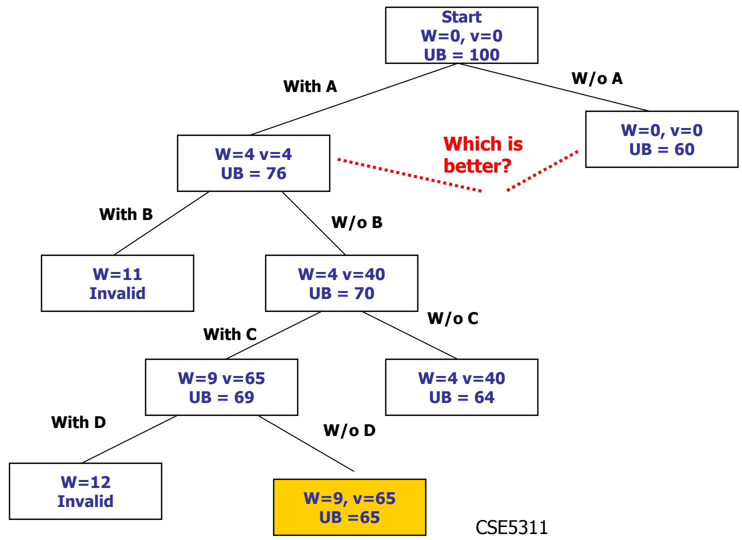
- UB = $v + (W-v)(v_{i+1}/w_{i+1})$
- When we start v = 0

W = 10

Item	Weight	Value	Value/ weight
A	4	$40	10
B	7	$42	6
C	5	$25	5
D	3	$12	4

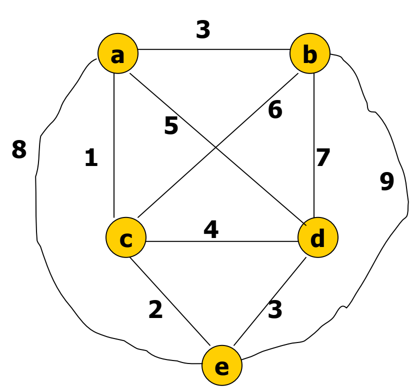
Knapsack

$UB = v + (W-v)(v_{i+1}/w_{i+1})$



Traveling Salesperson Problem

- LB = $\sum$ (distance to two nearest cities)/2
- $\sum$ over all cities



Problems

Item	Weight	Value	Value/ weight	
A	10	$100		
B	7	$63		
C	8	$56		
D	4	$12