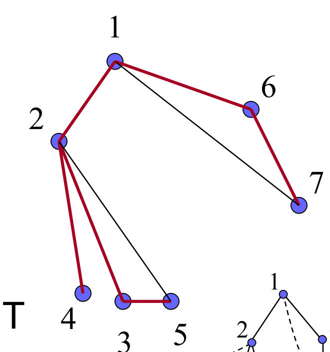
Graphs & Games

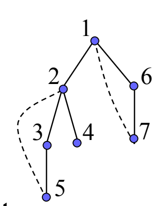
EECS 477

Lecture 18, 11/14/2002

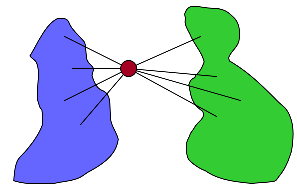
DFS properties



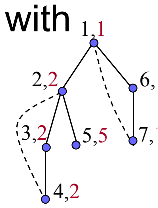
- Undirected graphs
  - Takes $\Theta(|E|+|V|)$ time
  - Builds a spanning tree T
    - Example
  - If not connected get a forest
  - Edges not in T connect a node to its ancestor (cannot cross to another branch)
  - Nodes of T indexed in pre-order (*prenum*)
    - Of course, depends on the starting node



Articulation points



- A node *v* of a connected graph
  - – is an articulation point if deleting it with adjacent edges makes the graph disconnected



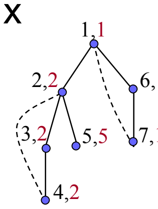
- Find them
- Define highest[v] = prenum of a highest node that can be reached going down the tree and at most one dashed link up

Articulation points

- (non-root) Node v is an articulation point

  if and only if it has at least one child x such that `highest[x]>=prenum[v]`

  - Indeed then subtree rooted at x will be separated from the rest of the graph
  - Root is articulated if it has more than one child
  - highest[v] = min(prenum[v], prenum[w], highest[u]) over all w's connected to v by dashed line and all children u
    - » (this is how we compute highest values)

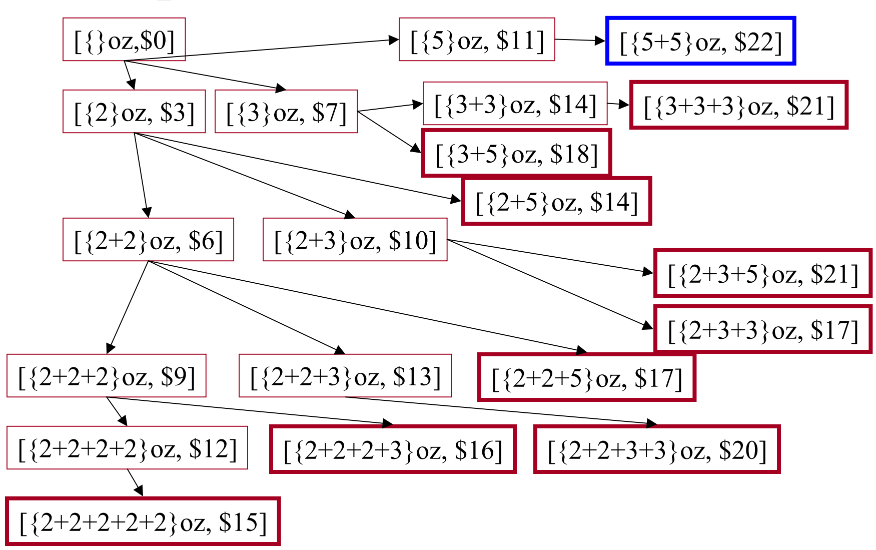


Backtracking

- Exploring implicit graph
  - similar to DFS in directed graph
- Solution consists of parts
  - Choice which to add
  - Knapsack: N types of objects
    - e.g. {(2oz, $3) (3oz, $4), (5oz, $10)}
    - W = 10
    - [{},0] – root of the tree
    - [{2,5},$13], etc.

Knapsack

{(2oz, $3) (3oz, $7), (5oz, $11)}, W=10

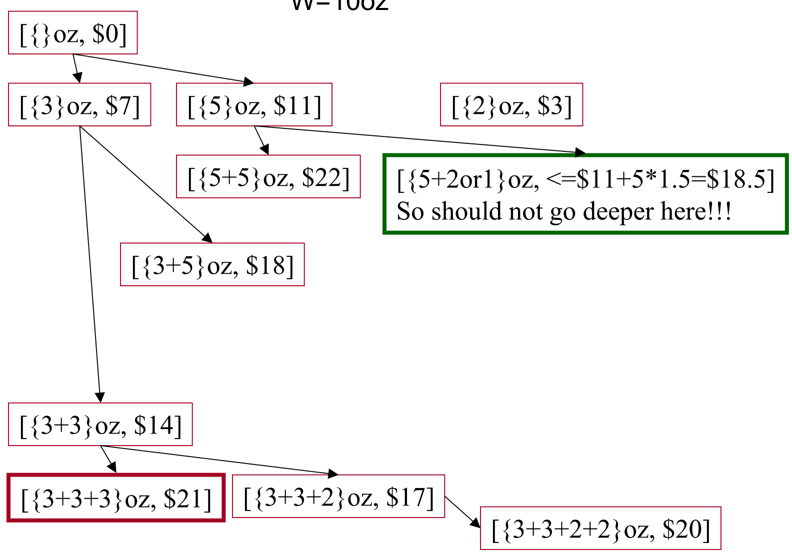


Knapsack

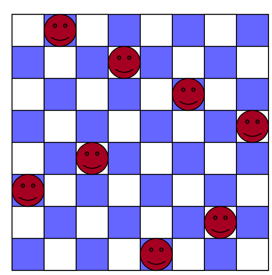
$2.33 $2.2 $1.5 $1

{(3oz, $7), (5oz, $11), (2oz, $3), (1oz, $1)}

W=10oz



Eight queens problem



- No threatening
- Solutions:
  - C(64,8) approx. 4 billions
  - Vector of 8 numbers 8^8 approx 16 millions
  - Permutations 8! = 40,320
  - Backtracking
    - DFS: tree of k-promising vectors (size 2057)
    - One queen at a time
    - Check right away – only the lastly added queen

Branch and bound

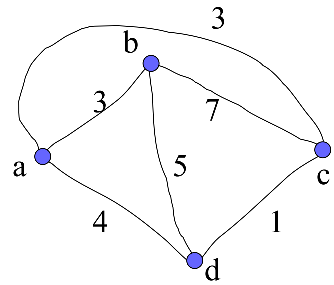
- Looking for an optimal solution
  - Use bounds to prune the search tree
  - DFS or BFS
  - Example: assignment
    - Matrix Cost[x,y]
    - Minimize Σ_x Cost[x, a[x]] where a[x] is the assignment and a[x]!=a[y] when x!=y
    - Assign jobs with least costs one per worker

Assignment

- Branch and bound
  - minimize Σ_x Cost[x, a[x]]
- Cost: diagonal, first A, then B,…

	job1	job2	job3	job4
A	8	12	3	10
B	11	7	8	9
C	2	14	9	4
D	12	5	12	7

TSP



- Branch and bound
- Need a lower bound
  - How good can a tour be?
  - (sum of min edge pair costs) / 2

  a:3+3, b:3+5, c:3+1, d:4+1.

  Cost >= (6+8+4+5)/2 = 23/2 = 11.5

  Cost[abdca] = 12. That must be optimal!
- Exclude/include edges one by one
  - That gives constraints

Minimax principle

- Two player games
- Games with values
  - Win/lose money
  - Chess/checkers: evaluation function
  - Player Max and player Min
  - Tree of variants: alternate min and max
    - Start from the leaves
    - Alpha-beta pruning
      - Not covered here