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 $\text{highest}[v] = \text{prenum}$ 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 $\text{highest}[x] \geq \text{prenum}[v]$
  - 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
  - $\text{highest}[v] = \min(\text{prenum}[v], \text{prenum}[w], \text{highest}[u])$ 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

- [{}, oz, $0] → [{2oz, oz, $3] → [{3oz, oz, $7} → [{5oz, oz, $11} → [{5+5oz, oz, $22}]

- [{2oz, oz, $6} → [{2+2oz, oz, $9] → [{2+2+2oz, oz, $12] → [{2+2+2+2oz, oz, $15}]

- [{2+3oz, oz, $10} → [{2+2+3oz, oz, $13} → [{2+2+2+3oz, oz, $16} → [{2+2+2+2+3oz, oz, $20}]

- [{3+5oz, oz, $18} → [{2+5oz, oz, $14} → [{2+3+3oz, oz, $17} → [{2+3+5oz, oz, $21}]

- [{3+3+3oz, oz, $14} → [{3+3+3oz, oz, $21} → [{3+3+3+3oz, oz, $21}]

- [{5oz, oz, $11} → [{5+5oz, oz, $22} → [{5+5oz, oz, $22} → [{5+5oz, oz, $22}
Knapsack

\{(3oz, $7), (5oz, $11), (2oz, $3), (1oz, $1)\}
\[ W=10oz \]

\$2.33 \quad \$2.2 \quad \$1.5 \quad \$1 \]

- \{3\}oz, $7$
- \{5\}oz, $11$
- \{2\}oz, $3$
- \{5+5\}oz, $22$
- \{3+5\}oz, $18$
- \{3+3\}oz, $14$
- \{3+3+3\}oz, $21$
- \{3+3+2\}oz, $17$
- \{3+3+2\}oz, $20$

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 $\sum x \text{ Cost}[x, a[x]]$ 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 $\sum x \text{ Cost}[x, a[x]]$
- Cost: diagonal, first A, then B,...

<table>
<thead>
<tr>
<th></th>
<th>job1</th>
<th>job2</th>
<th>job3</th>
<th>job4</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>8</td>
<td>12</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>B</td>
<td>11</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>14</td>
<td>9</td>
<td>4</td>
</tr>
<tr>
<td>D</td>
<td>12</td>
<td>5</td>
<td>12</td>
<td>7</td>
</tr>
</tbody>
</table>
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