Traveling Salesperson Problem

Given: n cities, and the pairwise distances d_{ij}

Goal: find shortest tour that visits every city at least once

Traveling Salesperson Problem

Given: n cities, and the pairwise distances d_{ij}

Goal: find shortest tour that visits every city at least once

Brute force search algorithm: $n! \sim 2^{n\log n}$ time.

Traveling Salesperson Problem

Given: n cities, and the pairwise distances d_{ij}

Goal: find shortest tour that visits every city at least once

Brute force search: $n! \sim 2^{n\log n}$ time.

Subproblems: T(v,S) - length of shortest tour that visits all cities of the set S and ends at v.

Given: n cities, and the pairwise distances d_{ij}

Goal: find shortest tour that visits every city at least once

Subproblems: $T(v,S)$ - length of shortest tour that visits all cities of the set S and ends at v.

Observation:
if shortest tour for $T(v,S)$ visits city u right before v, then

$$T(v,S) = T(u,S-v) + d_{uv}$$

Given: n cities, and the pairwise distances d_{ij}

Goal: find shortest tour that visits every city at least once

Subproblems: T(v,S) - length of shortest tour that visits all cities of the set S and ends at v.

Algorithm:

```
for v=1,...,n
    set T(v,{v}) = 0
for k=2,...,n
    for all sets of cities S, |S|=k
        for all v in S
            set T(v,S) = min        T(u,S-v)+d_uv
                       u in S-v
```

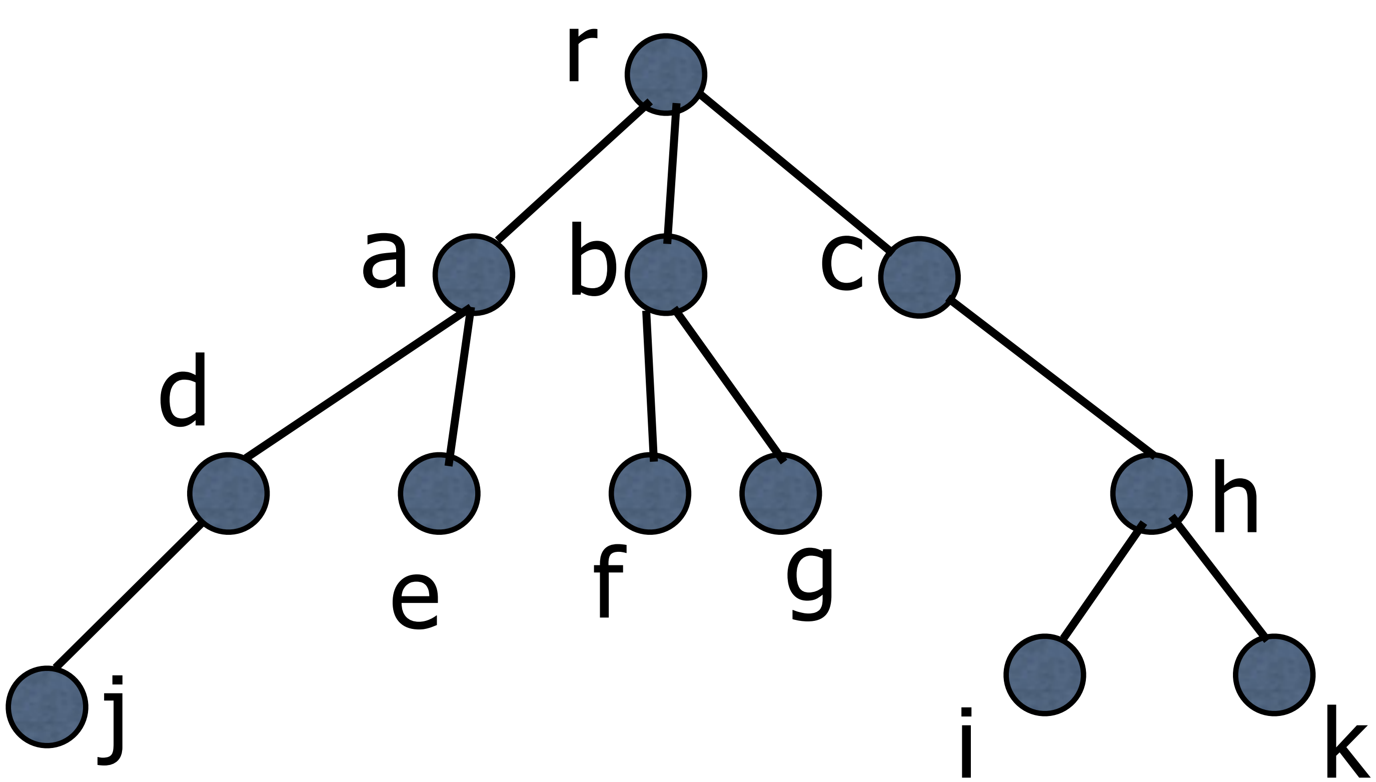
Running time

$O(n^2 \, 2^n)$

Vertex Cover on Acyclic Graphs

Given: A tree

Goal: find smallest vertex cover (vertices that touch all edges)

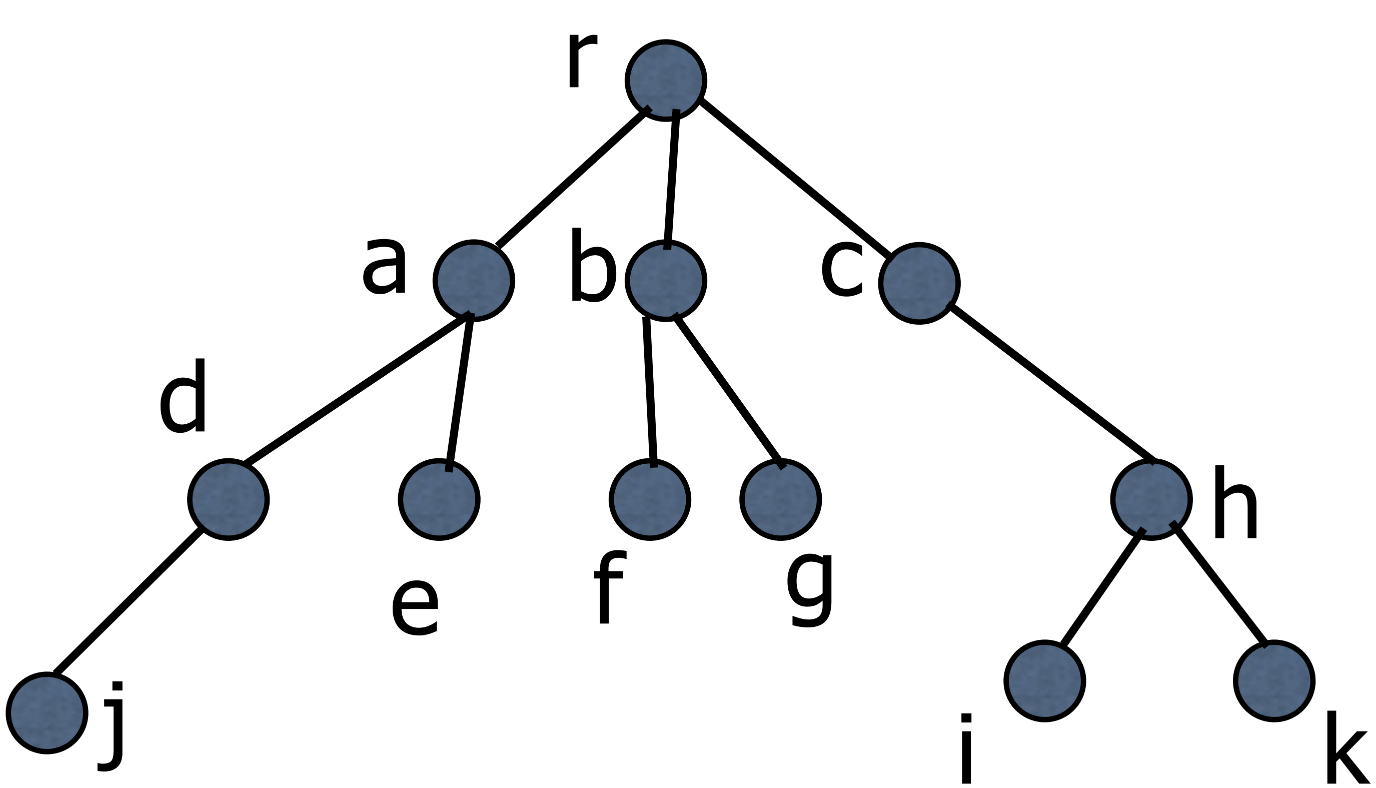


Vertex Cover on Acyclic Graphs

Given: A tree

Goal: find smallest vertex cover (vertices that touch all edges)

Subproblems: V(r) - size of vertex cover at subtree rooted at r.

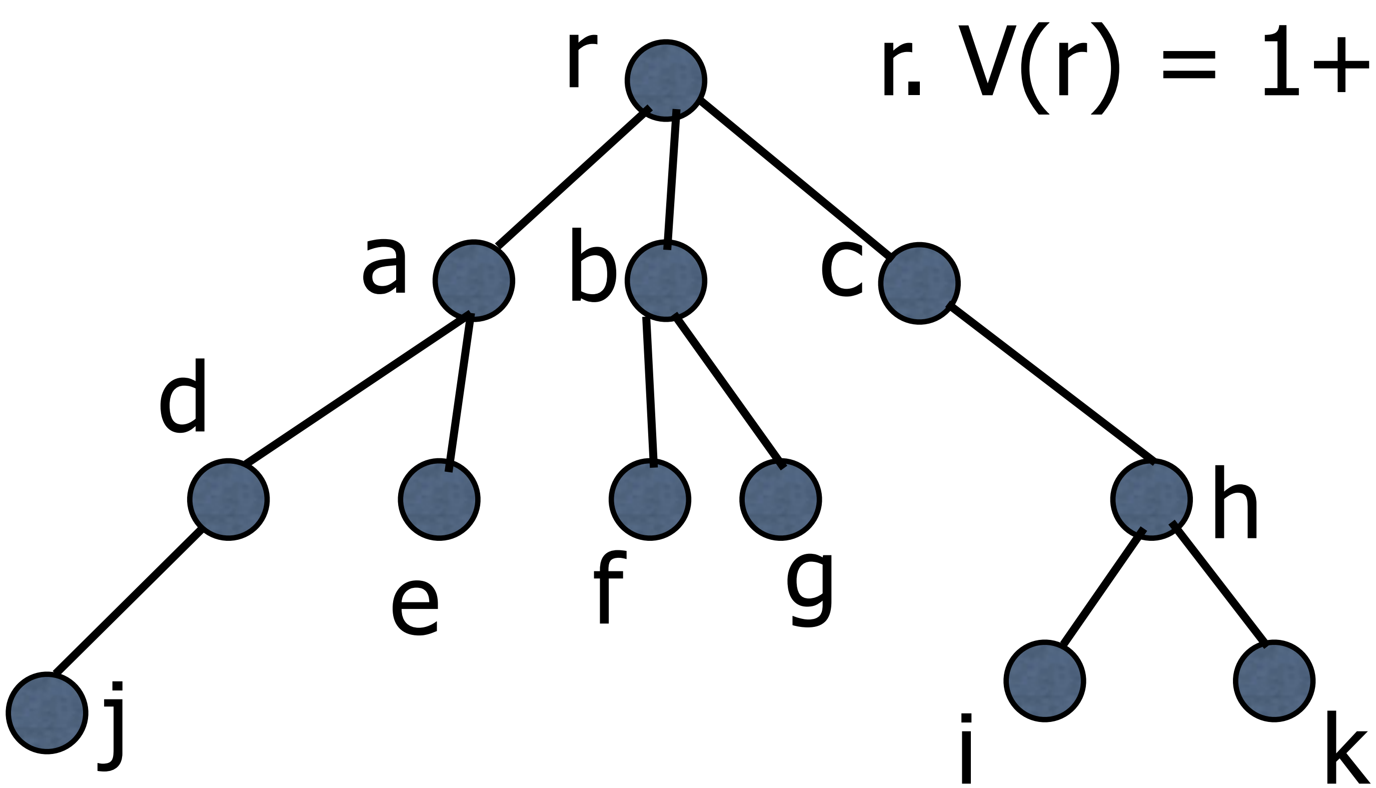


Vertex Cover on Acyclic Graphs

Subproblems: V(r) - size of vertex cover at subtree rooted at r.

Case 1: Cover realizing V(r) does not contain r. Then it must contain children(r).
V(r)= #children(r) + sum over grandchilren g V(g)

Case 2: Cover realizing V(r) does contain r. V(r) = 1+sum over children c V(c)



Vertex Cover on Acyclic Graphs

Subproblems: V(r) - size of vertex cover at subtree rooted at r.

Case 1: Cover realizing V(r) does not contain r. Then it must contain children(r).
V(r)= #children(r) + sum over grandchilren g V(g)

Case 2: Cover realizing V(r) does contain r.
V(r) = 1+sum over children c V(c)

Rough Algorithm:

Running time
O(n)

For each vertex r, in decreasing order of depth, set

$$V(r) = \min\{\#children(r)+\sum_{g,\ \text{grandchild of } r} V(g),\ 1+ \sum_{c,\ \text{child of } r} V(c)\}$$

Chain Matrix Multiplication

Given: n matrices $M_1, M_2, \ldots, M_n$

Goal: compute product $M_1, M_2, \ldots, M_n$ (in what order should we multiply?)

Example: To compute VWXYZ we could multiply V((WX)(YZ)) or (V(W(XY)))Z or ...

Basic operations: multiplying (a by b) matrix with (b by c) matrix gives (a by c) matrix in abc time.

Chain Matrix Multiplication

Given: n matrices $M_1, M_2, \ldots, M_n$

Goal: compute product $M_1, M_2, \ldots, M_n$ (in what order should we multiply?)

Example: To compute VWXYZ we could multiply V((WX)(YZ)) or (V(W(XY)))Z or ...

Basic operations: multiplying (a by b) matrix with (b by c) matrix gives (a by c) matrix in abc time.

Subproblems: C(i,j) - time to compute $M_i M_{i+1} \ldots M_j$

Given: n matrices $M_1,\ldots,M_n$, i'th matrix of size (m_i by m_{i+1})

Goal: compute product $M_1,M_2,\ldots,M_n$ (in what order should we multiply?)

Basic operations: multiplying (a by b) matrix with (b by c) matrix gives (a by c) matrix in abc time.

Subproblems: C(i,j) - time to compute $M_iM_{i+1}\ldots M_j$

Observation: If the final multiplication in optimal solution is between $(M_i\ldots M_k)(M_{k+1}\ldots M_j)$, then
$C(i,j) = C(i,k)+C(k,j)+ n_in_{k+1}n_j$.

Basic operations: multiplying (a by b) matrix with (b by c) matrix gives (a by c) matrix in abc time.

Subproblems: $C(i,j)$ - time to compute $M_i M_{i+1} \ldots M_j$

Observation: If the final multiplication in optimal solution is between $(M_i \ldots M_k)(M_{k+1} \ldots M_j)$, then
$C(i,j) = C(i,k)+C(k+1,j)+ m_i m_{k+1} m_j$.

Algorithm:

Running time
$O(n^3)$

for $i=1,2,\ldots,n-1$, set $C(i,i)=0$

for $s=1,2,\ldots,n-1$, $i=1,\ldots,n-1$
set $C(i,i+s)=\min_{i<k<s} C(i,k)+C(k+1,i+s)+m_i m_{k+1} m_{i+s}$

Common Subproblems

- Opt(i) - Opt solution using $x_1,..,x_i$. (eg LIS, longest path).
- Opt(i,j) - Opt solution using $x_i,...,x_j$. (eg RNA)
- Opt(i,j) - Opt solution using $x_1,...,x_i$ and $y_1,...,y_j$. (eg Edit distance)
- Opt(r) - Opt solution using subtree rooted at r. (eg Vertex cover on trees).