Dynamic Programming - III

DP on a tree

DP on a tree

- Input : A weighted rooted tree, T
- Output : Maximum weight Independent Set of T

DP on a tree

- Sub-problem :

A[v] : Maximum weight Independent Set of the subtree rooted at v

DP on a tree

- Sub-problem :

A[v] : Maximum weight Independent Set of the subtree rooted at v

Let root be r. Then A[r] gives the final solution.

DP on a tree

- Sub-problem :

A[v] : Maximum weight Independent Set of the subtree rooted at v

- v is not in the solution :

- v is in the solution :

DP on a tree

$$A[v] = \max \left[1 + \sum_{w:\text{grandchild of } v} A[w], \sum_{u:\text{child of } v} A[u] \right]$$

Proof of correctness:

Proof of correctness:

By induction on height on the subtree

Bottom-up Implementation

Running time:

DP on a tree

- Input : A weighted rooted tree, T
- Output : Minimum weight Dominating Set of T

DP on a tree

- Input : A weighted rooted tree, T
- Output : Minimum weight Dominating Set of T

A dominating set of a graph G is a subset D of vertices such that every vertex either is in D or has at least one neighbour in D

Minimum Dominating Set Problem

- Input : A path, P
- Output : Minimum Dominating Set of P

A dominating set of a graph G is a subset D of vertices such that every vertex either is in D or has at least one neighbour in D

Minimum Dominating Set Problem

- Input : A weighted path, P
- Output : Minimum weight Dominating Set of P

A dominating set of a graph G is a subset D of vertices such that every vertex either is in D or has at least one neighbour in D

Minimum Dominating Set Problem

- OPT[i] : value of the minimum dominating set for the sub-path containing first i vertices

Minimum Dominating Set Problem

- OPT[i] : value of the minimum dominating set for the sub-path containing first i vertices
- i is not in the solution :
- i is in the solution :

Minimum Dominating Set Problem

- OPT[i] : value of the minimum dominating set for the sub-path containing first i vertices
- i is not in the solution : i-1 must be present in the solution
- i is in the solution : i-1 may or may not be in the solution.

Minimum Dominating Set Problem

- OPT[i] : value of the minimum dominating set for the sub-path containing first i vertices

- OPT_with[i] : value of the minimum dominating set that contains i for the sub-path containing first i vertices

- OPT_dc[i] : value of the minimum dominating set that may or may not dominate i for the sub-path containing first i vertices

Minimum Dominating Set Problem

- OPT[i] = min (OPT_with[i], OPT_with[i-1])

Minimum Dominating Set Problem

- OPT[i] = min (OPT_with[i], OPT_with[i-1])
- OPT_with[i] = w(i) + OPT_dc[i-1]

Minimum Dominating Set Problem

- OPT[i] = min (OPT_with[i], OPT_with[i-1])
- OPT_with[i] = w(i) + OPT_dc[i-1]
- OPT_dc[i] = min (OPT[i-1], OPT_with[i])

Minimum Dominating Set Problem

- Running Time ?

Minimum Dominating Set Problem on a tree

- OPT[v] : value of the minimum dominating set for the sub-tree rooted at v

- OPT_with[v] : value of the minimum dominating set that contains v for the sub-tree rooted at v

- OPT_without[v] : value of the minimum dominating set that does not contain v for the sub-tree rooted at v

- OPT_dc[v] : value of the minimum dominating set that may or may not dominate v for the sub-tree rooted at v

Minimum Dominating Set Problem on a tree

- OPT[v] = min(OPT_with[i], opt_without[i])

Minimum Dominating Set Problem on a tree

- OPT[v] = min(OPT_with[i], opt_without[i])
- $$\text{OPT\_with}[v] = w(v) + \sum_{u-\text{child of } v} \text{OPT\_dc}[u]$$

Minimum Dominating Set Problem
on a tree

- OPT[v] = min(OPT_with[i], opt_without[i])
- $$\text{OPT\_with}[v] = w(v) + \sum_{u-child\ of\ v} \text{OPT\_dc}[u]$$
- OPT_without[v]

$$= \min_{u-child\ of\ v} \left(\text{OPT\_with}(u) + \sum_{w-child\ of\ v, w \neq u} \text{OPT}[w] \right)$$

Minimum Dominating Set Problem on a tree

- OPT[v] = min(OPT_with[i], opt_without[i])

- OPT_with[v] = w(v) + $\sum_{u-child\ of\ v}$ OPT_dc[u]

- OPT_without[v]

$$= \min_{u-child\ of\ v} \left(\text{OPT\_with}(u) + \sum_{w-child\ of\ v, w \neq u} \text{OPT}[w] \right)$$

- OPT_dc[v] = min($\sum_{u-child\ of\ v}$ OPT[u], OPT_with[v])