20C041

Dynamic Programming

Forward Approach

$$Cost(i,j) = Min[c(j,r) + Cost(i+1, r)]$$

stage vertex vertex stage vertex in next stage

* Destination to Source to solve $d[j] = r$
* To find shortest distance source to destination

Backward Approach

$$Cost(i,j) = Min\{c(r,j) + Cost(i-1, r)\}$$

stage vertex vertex stage Previous vertex in ~~next~~ stage

$d[j] = r$

* Source to Destination
* To find shortest distance destination to Source.

0/1 Knapsack

Tabular Method.

$$T[i,j] = Max\{T[i-1,j], P_i + T[i-1, j-w_i]\}$$

if $W_i >= j$

if $(T(i,k) \neq T[i-1,k])$

$i = i-1$ & $k = k - w_i$;

Else

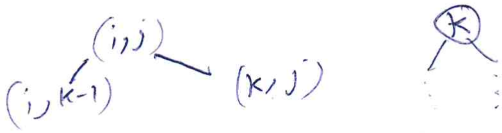
$i = i-1$

OBST

$$W(i,j) = \sum_{k=i+1}^{j} p_k + \sum_{k=i}^{j} q_k \quad ; i<j$$

$j-i = 0, 1, 2, 3, 4 \ldots\ldots$ or $W(i,j) = p_j + q_j + W(i,j-1)$

$$C(i,j) = \min_{i<k\leq j} \{ C(i,k-1) + C(k,j) \} + W_{ij}$$



$$r(i,j) = k$$

TRAVVILING SALESPERSON PROBLEM

$$g(i,S) = \min_{j \in S} [c(i,j) + g(j, S-\{j\})]$$

$P[i,S] = S$

All pairs Shortest Path Problem
(Floyd Warshell Algo)

$$A^k(i,j) = A^{k-1}(i,k) + A^{k-1}(k,j),\ A^{k-1}(i,j)$$

Single Source Shortest path Using DP
(Bellman) Ford Algorithm)

$$dist^k[u] = \min\{ dist^{k-1}[u], \min\{ dist^{k-1}[i] + cost[i,u] \} \}$$

Dynamic

* Forward / backward

$TC = \theta(|V| + |E|) + |k|$ ← find minimum cost path

↑ stage & relaxation

$= \theta(|V| + |E|)$

* 0/1 knapsack

Set Method & Tabular Method

$TC = O(nm)$ m→capacity of knapsack

* TSP

$TC = N \times 2^N$

* All pair shortest path (Floyd)

$TC = \theta(n^3)$

* Single Source (Bellman ford)

$TC = (|E| * n-1) \Rightarrow O((n-1) * (n-1)) = O(n^2)$

GREEDY

* Fractional Knapsack

Time Complexity

a) minimum time to sort the array : $O(n \log n)$

b) Time required to choose the feasible set : $\sum_{1}^{n} 1 = n = O(n)$

$TC = O(n \log n) + O(n) = O(n \log n)$

* Krushal

Construction of heap $O(|E|)$

sorting the edge based on weights :

$TC = O(|E| \log |E|)$ $E \to$ Edge set of G

* PRIM

$TC = O(n^2)$ OR $O(V^2)$ where $n = V =$ vertices

TC can be reduced using Binary heap

* Dijkstra's Algorithm

$O((n + |E|) \log n)$